

Universidad de las Ciencias Informáticas

Facultad 3

Centro de Informatización de Entidades



**“Solución informática para evaluar los gastos de funcionamiento interno en el Sistema de Planificación de Actividades SIPAC”**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

Autor: Luis Angel Serrano Carrión

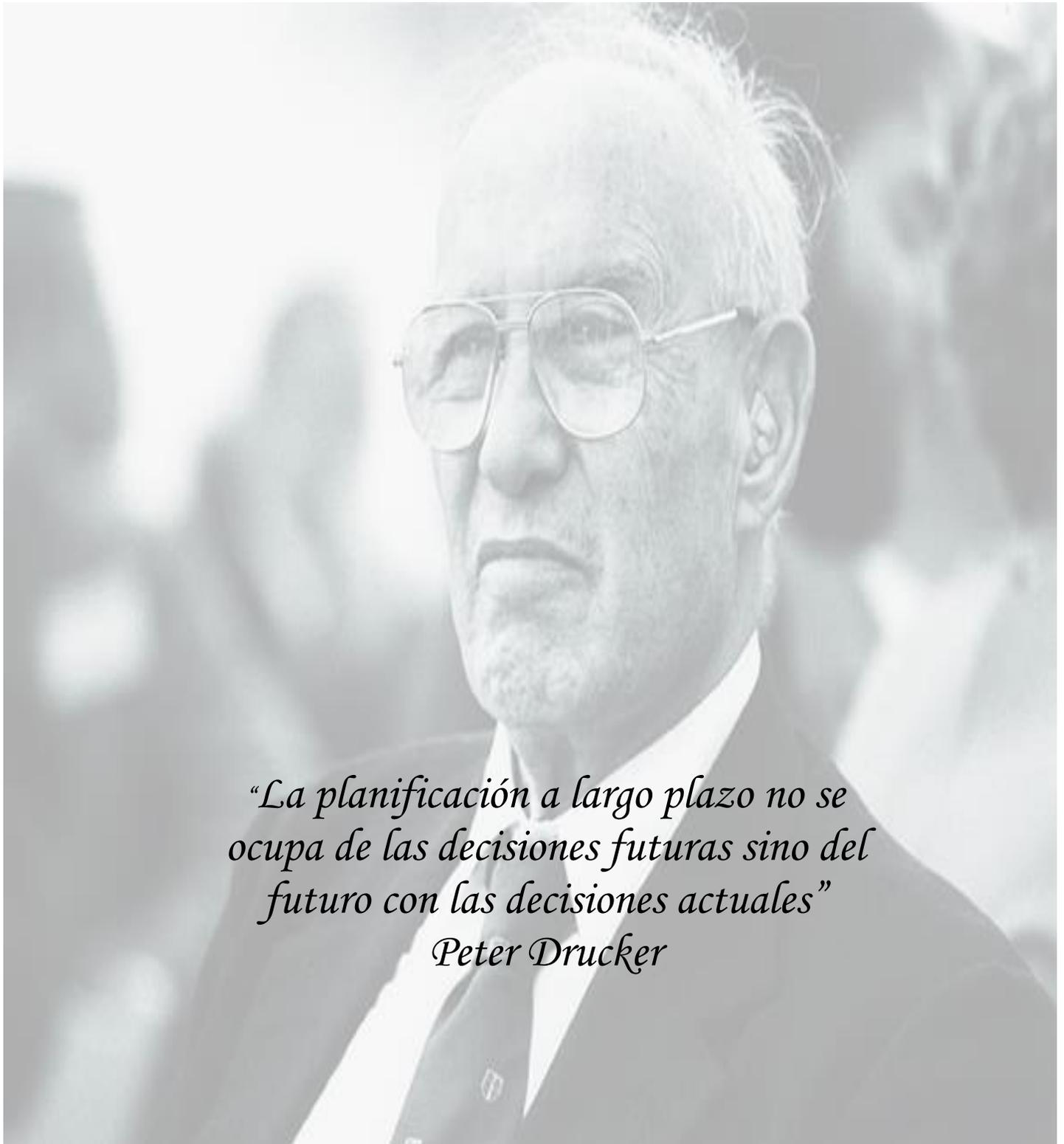
Tutor(es): Ing. Mairelys Fernández González

Ing. Ailia Parra Fernández

Ing. Yadelis Trujillo Martínez

La Habana

Junio 2015



*“La planificación a largo plazo no se ocupa de las decisiones futuras sino del futuro con las decisiones actuales”*

*Peter Drucker*

## **DECLARACIÓN DE AUTORÍA**

Declaro ser autor de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Luis Angel Serrano Carrión**

**Autor**

---

**Ing. Yadelis Trujillo Martínez**

**Tutor**

---

**Ing. Ailia Parra Fernández**

**Tutor**

---

**Ing. Mairelys Fernández González**

**Tutor**

## **AGRADECIMIENTOS**

*A mis padres por hacer de mí el hombre que soy a pesar de todas las adversidades.*

*A mi esposa por su apoyo incondicional, por su comprensión y por lo que representa dentro de mi vida.*

*A mis abuelos y tíos por su confianza, su apoyo y por creer siempre en mí.*

*A Migdalia por toda su ayuda siendo otra madre para mí.*

*A mi hermano por estar ahí cuando lo necesito y por todo lo que hemos vivido juntos.*

*A mis amigos Leosbel, Carlos, Lorena, Lienzt, Yadiesqui, Rinat, Ibanie, Eduardo, Carlos José, Leonel por su ayuda, sus consejos y su apoyo.*

*A mis tutoras por haberme apoyado y guiado en el desarrollo de este trabajo.*

*A los profesionales de proyecto SIPAC en especial a Yordi y Teresa.*

*A los profesores Calos Abel y Dayannis por su ayuda y sus consejos para realizar este trabajo.*

*A mis compañeros de cuarto, a mi grupo y a todas la personas que han hecho esto posible.*

**DEDICATORIA**

*Dedico este trabajo a mi familia y en especial a mi niña Leslie que espero un día se sienta orgullosa de su papá.*

## **RESUMEN**

El presente trabajo tiene como objetivo desarrollar una solución informática para evaluar el gasto en las actividades de aseguramiento interno en el Sistema de Planificación de Actividades, de manera que facilite el registro, seguimiento y control de los recursos en la planeación operativa. Los principales resultados obtenidos referencian todos los artefactos de cada una de las disciplinas de la fase de ejecución del modelo de desarrollo de la Universidad de las Ciencias Informáticas. La solución informática permite asignar los recursos a las actividades, evaluar y controlar la ejecución de las mismas, lográndose una comparación entre lo planificado y lo real ejecutado, lo que contribuye a evaluar la eficiencia en el logro de los objetivos, que propicie el desarrollo organizacional, alineado a la implementación efectiva de los lineamientos de la Política Económica y Social del Partido y la Revolución.

**Palabras clave:** actividades de aseguramiento interno; planificación; recursos.

---

**ÍNDICE**

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Conceptos generales .....	5
1.1.1 Planificación .....	5
1.1.2 Gastos .....	5
1.1.3 Funcionamiento interno .....	5
1.1.4 Actividades de funcionamiento interno .....	6
1.1.5 Recursos materiales.....	6
1.1.6 Presupuesto .....	7
1.2 Recursos materiales para el desarrollo de la solución propuesta .....	7
1.3 Estado del arte.....	9
1.3.1 Sistemas de planificación existentes a nivel mundial .....	9
1.3.2 Resolución del Ministerio de Finanzas y Precios.....	11
1.4 Valoración de los sistemas de gestión estudiados.....	13
1.5 Metodología de desarrollo.....	14
1.6 Patrones de diseño .....	15
1.6.1 Patrones GRASP.....	15
1.6.2 Patrones GOF .....	16
1.7 Lenguajes, tecnologías y herramientas propuestas para el desarrollo de la solución .....	16
1.7.1 Lenguajes de modelado .....	17
1.7.2 Lenguajes de desarrollo del lado del cliente.....	17
1.7.3 Lenguajes de desarrollo del lado del servidor .....	18
1.7.4 Tecnologías y herramientas de desarrollo.....	18
CAPÍTULO II: ANÁLISIS Y DISEÑO .....	22
2.1 Modelado del negocio .....	22
2.1.1 Diagrama de procesos de negocio .....	22
2.1.2 Especificación del proceso de negocio: Establecer plan .....	24

---

2.1.3 Modelo conceptual .....	26
2.2 Requisitos .....	28
2.2.1 Técnicas de captura de requisitos .....	28
2.2.2 Requisitos funcionales.....	28
2.2.3 Especificación de requisitos funcionales .....	30
2.2.4 Validación de requisitos.....	31
2.2.5 Requisitos no funcionales.....	31
2.3 Modelado de la solución propuesta .....	33
2.3.1 Diagrama de clases del diseño.....	33
2.3.2 Diagrama de secuencia.....	35
2.3.3 Modelo de datos .....	36
2.3.4 Diagrama de componentes .....	38
2.4 Patrones de diseño utilizados .....	39
2.4.1 Patrones GRASP .....	39
2.4.2 Patrones GOF .....	39
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS .....	41
3.1 Validación del diseño propuesto .....	41
3.2 Implementación de la solución propuesta .....	43
3.2.1 Estándares de codificación.....	43
3.2.2 Evaluación de los gastos para las actividades de aseguramiento interno en SIPAC ...	45
3.3 Pruebas de <i>software</i> .....	49
3.4 Resultados de las pruebas aplicadas .....	54
CONCLUSIONES.....	56
RECOMENDACIONES .....	57
REFERENCIAS BIBLIOGRÁFICAS .....	58
BIBLIOGRAFÍA CONSULTADA .....	62
GLOSARIOS DE TÉRMINOS.....	63
ANEXOS.....	64

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Desglose de los recursos materiales y sus gastos asociados .....	8
<b>Tabla 2:</b> Comparación de los sistemas .....	13
<b>Tabla 3:</b> Especificación del proceso de negocio: Establecer plan.....	24
<b>Tabla 4:</b> Especificación del requisito: Registrar gastos de elementos de oficina .....	30
<b>Tabla 5:</b> Caso de prueba para el camino básico # 1.....	52
<b>Tabla 6:</b> Caso de prueba de caja negra para validar el requisito funcional Registrar gastos de elementos de oficina.....	53
<b>Tabla 7:</b> Material de medición de la métrica TOC.....	64
<b>Tabla 8:</b> Material de medición de la métrica RC. ....	64
<b>Tabla 9:</b> Caso de prueba para el camino básico # 2.....	65
<b>Tabla 10:</b> Juego de datos a probar para el caso de prueba del requisito funcional Registrar gastos de elementos de oficina.....	66

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Diagrama de proceso de negocio Establecer plan. ....	23
<b>Figura 2:</b> Modelo conceptual de los gastos de funcionamiento interno para SIPAC. ....	27
<b>Figura 3:</b> Vista del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar los gastos de las actividades de aseguramiento interno. ....	33
<b>Figura 4:</b> Controlador del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar el gasto de las actividades de aseguramiento interno. ....	34
<b>Figura 5:</b> Modelo del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar el gasto de las actividades de aseguramiento interno. ....	34
<b>Figura 6:</b> Diagrama de secuencia del requisito funcional Registrar gastos de los elementos de oficina. ....	35
<b>Figura 7:</b> Modelo de datos para la evaluación de las actividades de aseguramiento interno. ....	37
<b>Figura 8:</b> Diagrama del componente planeación en el SIPAC. ....	38
<b>Figura 9:</b> Representación en porcentaje de la cantidad de clases y el número de procedimientos que contienen. ....	42
<b>Figura 10:</b> Representación en porcentaje de las asociaciones de uso por cantidad de clases. ....	43
<b>Figura 11:</b> Adicionar actividad. ....	45
<b>Figura 12:</b> Registrar gastos de bolígrafos. ....	46
<b>Figura 13:</b> Definir presupuesto al plan. ....	46
<b>Figura 14:</b> Analizar plan. ....	47
<b>Figura 15:</b> Evaluación de los gastos de funcionamiento interno. ....	48
<b>Figura 16:</b> Grafo de flujo asociado al procedimiento obtenerRecursosSegunActividad. ....	51
<b>Figura 17:</b> No conformidades detectadas durante las iteraciones de pruebas realizadas. ....	55
<b>Figura 18:</b> Fragmento de código del algoritmo obtenerRecursosSegunActividad. ....	65

## **INTRODUCCIÓN**

El término planificar se atribuye al establecimiento de objetivos con el fin de alcanzar determinados resultados y elegir un futuro curso de acciones para lograrlos. La planificación de objetivos y actividades en Cuba ha venido madurando a partir de la experiencia de la aplicación del perfeccionamiento empresarial y la evaluación de la ejecución de los planes en cada entidad del país. Como resultado de esta continua evaluación surge la necesidad de crear un órgano de dirección del gobierno, capaz de dirigir la planificación, así como el documento rector para encauzar este proceso; que aseguren el cumplimiento de los objetivos, la integralidad en los planes económicos y la adecuada vinculación con la planificación de las actividades. Surge así la Instrucción no.1 del Presidente de los Consejos de Estado y de Ministros para la Planificación de los objetivos y actividades en los órganos, Organismos de la Administración Central de Estado, entidades nacionales y Administraciones locales del Poder Popular.

La Instrucción no.1 establece las políticas generales a seguir para la elaboración de los objetivos y planes de actividades en los diferentes niveles de dirección, que den cumplimiento a los acuerdos y resoluciones aprobadas en el VI Congreso del Partido Comunista de Cuba, las decisiones de la Asamblea Nacional del Poder Popular, el Consejo de Ministros y la actualización de los planes de la economía (1).

El Sistema para la Planificación de Actividades (SIPAC), desarrollado en el Centro de Informatización de Entidades (CEIGE) de la Universidad de las Ciencias Informáticas (UCI), constituye la informatización de la Instrucción no.1. SIPAC interrelaciona objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades como parte de la planificación a corto, mediano y largo plazo (2). Cuenta con varios módulos y componentes (Configuración, Planeación, Notificaciones y Recuperaciones). Independientemente de las funcionalidades que brinda el sistema, no cuenta con ningún método o procedimiento formal para evaluar los gastos de funcionamiento interno, proceso que en la actualidad se realiza de forma manual o con hojas de cálculo y procesadores de texto, lo que trae como consecuencia el aumento del margen de error dado por la manipulación de los datos de una o varias personas, lo que crea además un escenario propicio a ilegalidades dificultando al mismo tiempo el control de esta información.

Las formas actuales de llevar a cabo este proceso, no cumplen con las necesidades de mantener actualizado todo el registro de gastos internos en la organización para su funcionamiento, restan

tiempo a las personas encargadas de los procesos relacionados y no son eficientes para la realización de las actividades asociadas a los mismos.

Por lo anteriormente expuesto se plantea como **problema a resolver**: ¿Cómo evaluar el gasto en las actividades de aseguramiento interno en la planeación operativa, de manera que facilite el registro, seguimiento y control de los recursos en SIPAC?

Se define como **objeto de estudio**: Evaluación de los gastos de funcionamiento interno en la planeación operativa.

La presente investigación se encuentra enmarcada en el **campo de acción**: Evaluación de los gastos de funcionamiento interno en la planeación operativa en SIPAC.

Para darle solución al problema planteado se tiene como **objetivo general**: Desarrollar la solución informática para evaluar el gasto en las actividades de aseguramiento interno en SIPAC, de manera que facilite el registro, seguimiento y control de la planeación operativa.

Para cumplir con el objetivo general se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación a partir de soluciones informáticas existentes relacionadas con el campo de acción, para identificar elementos afines con la solución que se desea desarrollar.
- Realizar el análisis y diseño de la solución informática que evalúa el gasto de las actividades de aseguramiento interno en la planeación operativa, para comprender mejor la propuesta de solución.
- Implementar la solución informática para evaluar el gasto de las actividades de aseguramiento interno en la planeación operativa, en SIPAC.
- Validar la solución propuesta mediante los métodos de pruebas de caja negra y caja blanca, para que la evaluación de los gastos de las actividades de aseguramiento interno cumpla con la calidad requerida.

#### **Tareas de la Investigación:**

1. Caracterización de los procesos relacionados con los gastos de funcionamiento interno en la planeación operativa.

2. Análisis de la arquitectura del SIPAC para conocer las características fundamentales, marcos de trabajo, herramientas y tecnologías definidas para el desarrollo, así como la estrategia de integración entre los diferentes componentes.
3. Estudio del proceso de desarrollo de *software* de la Universidad de las Ciencias Informáticas.
4. Estudio de los patrones de diseño.
5. Elaboración del modelo conceptual y el glosario de términos asociados a la solución.
6. Identificación, especificación y validación de los requisitos funcionales del *software*.
7. Diseño de los prototipos de interfaz de usuario.
8. Estudio y actualización del modelo de datos del SIPAC.
9. Elaboración del diseño de las clases asociado a la solución.
10. Validación del diseño aplicando las métricas adecuadas para esta etapa.
11. Implementación de los requisitos identificados.
12. Validación mediante la aplicación de pruebas de caja blanca y pruebas de caja negra.

Se define como **Idea a defender**: Si se desarrolla una solución que permita evaluar el gasto para las actividades de aseguramiento interno en SIPAC, se facilitará el registro, seguimiento y control de la planeación operativa.

Para el estudio preliminar de la investigación se utilizaron los siguientes métodos científicos:

**Métodos teóricos:**

- Análisis - síntesis: se utilizó para realizar el estudio y análisis de la documentación empleada para el desarrollo de la investigación, la formulación del problema y los objetivos.
- Histórico-lógico: se utilizó para realizar un estudio del comportamiento actual de los sistemas de gestión empresarial existentes en el mundo y determinar su influencia en el problema actual de la investigación.
- Modelación: se utilizó para modelar los requerimientos funcionales que se proponen en la solución, su relación con los procesos actuales y el diseño de los nuevos componentes visuales, las entidades relacionales y sus características.

**Métodos empíricos:**

- Entrevistas: se realizaron entrevistas a los usuarios con conocimientos suficientes sobre gastos de funcionamiento interno, con el objetivo de obtener información sobre las principales deficiencias e identificar las funcionalidades que se requieren para erradicar las mismas.

**Estructura del documento:**

**Capítulo I Fundamentación teórica:** se expone el estudio del estado del arte de los sistemas vinculados al objeto de estudio, así como la fundamentación de los conceptos asociados al dominio del problema a resolver. Se hace un estudio de la metodología de desarrollo propuesta y se caracterizan las herramientas y lenguajes utilizados para el desarrollo de la solución.

**Capítulo II Análisis y diseño:** se generan los principales artefactos del análisis y diseño de la solución propuesta, dando cumplimiento a las disciplinas definidas en la metodología de desarrollo, además se hace una especificación de los patrones utilizados para el diseño del sistema.

**Capítulo III Implementación y prueba:** se describe todo lo relacionado con la implementación de la solución propuesta, así como la validación de la misma empleando pruebas de caja blanca y pruebas de caja negra.

## **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA**

### **Introducción**

El presente capítulo estará centrado en la realización de un profundo análisis para obtener información actualizada de algunos sistemas de planificación existentes a nivel mundial y a partir de ahí conocer qué aspectos deben ser tomados en cuenta para la propuesta de solución. Será explicada la metodología de desarrollo a utilizar, la cual estará orientada a la solución. Se realizará un análisis de la arquitectura definida para el sistema y se especificarán las tecnologías y herramientas de desarrollo a utilizar durante la implementación, para facilitar el cumplimiento de los requisitos tanto técnicos como funcionales.

### **1.1 Conceptos generales**

#### **1.1.1 Planificación**

La planificación es una parte indispensable de la vida y de la sociedad que nos rodea. Las ideas concebidas con el fin de lograr una meta, alcanzar un propósito, de forma consciente o no, se traducen sencillamente en “planificación”. Uno de los exponentes más prominentes en materia de planificación, y considerado como el padre de la planificación en los tiempos actuales, James A. Finch, plantea que “la planificación no es más que el proceso de establecer metas y definir medios para alcanzar las mismas”. Por otra parte, el máster en administración y dirección de empresas Andrés M. Núñez Parada propone que la planificación “consiste en determinar las metas u objetivos a cumplir seleccionando las misiones, objetivos y acciones para alcanzarlos”. Partiendo de la esencia de las ideas anteriormente expuestas, este trabajo se acoge al concepto de que la planificación es una de las funciones básicas del proceso de gestión que permite establecer un nivel de estabilidad en las entidades y proporciona condiciones favorables ante los cambios avizorados o no en un futuro (3).

#### **1.1.2 Gastos**

Los gastos son decrementos en los beneficios económicos, producidos a lo largo del periodo contable, en forma de salidas o disminuciones del valor de los activos, o bien como surgimiento de obligaciones, que dan como resultado decrementos en el patrimonio neto o capital contable, y no están relacionados con las distribuciones de estos beneficios (4).

#### **1.1.3 Funcionamiento interno**

Funcionamiento es la acción y efecto de funcionar. Este verbo hace referencia a ejecutar las funciones que le son propias a algo o alguien, a aquello que marcha o resulta bien. Es el

comportamiento normal que un elemento tiene, comportamiento esperado para realizar una tarea específica (5). Interno es un adjetivo que puede utilizarse como sinónimo de interior. Se trata de aquello que está en la parte de adentro o que no tiene vistas al exterior (6).

Teniendo en cuenta los conceptos anteriores se puede concluir que el funcionamiento interno es la puesta en marcha de las funciones o actividades propias dentro de una organización para la concreción de ciertos fines. Es el comportamiento normal y esperado vinculado a la capacidad interna de una entidad para realizar tareas o acciones específicas que les permitan funcionar y conocer sus principales pilares.

#### **1.1.4 Actividades de funcionamiento interno**

El funcionamiento interno incluye actividades propias del organismo para su sistema de dirección, acciones de control interno, atención al hombre, aplicación de la política y el sistema de trabajo con los cuadros y sus reservas y el cumplimiento de las disposiciones vigentes (1).

#### **1.1.5 Recursos materiales**

Un recurso es un medio de cualquier clase que permite conseguir aquello que se pretende. Un material, por otra parte, es algo perteneciente o relativo a la materia. Los recursos materiales, en definitiva, son los medios físicos y concretos que ayudan a conseguir algún objetivo (7). Son aquellos recursos económicos de carácter tangible (8), que permiten a las organizaciones concretar sus objetivos.

**Recursos humanos (RRHH):** se denomina recursos humanos a las personas con las que una organización cuenta para desarrollar y ejecutar de manera correcta las acciones, actividades, labores y tareas que deben realizarse y que han sido solicitadas a dichas personas (9).

**Elementos de oficina:** incluyen diversos tipos de equipos y útiles que son necesarios para desempeñar determinadas acciones en la realización de las tareas de oficina (10).

**Equipos electrónicos:** consiste en una combinación de componentes electrónicos organizados en circuitos, destinados a controlar y aprovechar las señales eléctricas, que utilizan la electricidad para el almacenamiento, transporte o transformación de información (11).

**Materiales de empaque:** son materiales utilizados para la protección, el almacenamiento, la manipulación y el transporte de artículos o productos (12).

**Medios de transporte:** los medios de transporte son aquellos vehículos que sirven para transportar o trasladar personas u objetos de un lugar hasta otro (13).

**Medios de comunicación:** los medios de comunicación son instrumentos para informar y comunicar mensajes en versión textual, sonora, visual o audiovisual. Algunas veces son utilizados para comunicar de forma masiva a millones de personas, como es el caso de la televisión o los diarios impresos o digitales, y otras, para transmitir información a pequeños grupos sociales, como es el caso de los periódicos locales o institucionales (14).

#### **1.1.6 Presupuesto**

Un presupuesto es una previsión, proyección o estimación de gastos. Como tal, es un plan de acción cuyo objetivo es cumplir una meta prevista, expresada en valores y términos financieros. Es la cantidad de dinero necesaria para hacer frente a ciertos gastos. Los presupuestos son parte de la administración de las finanzas de familias, profesionales, empresas, organizaciones o países (15).

#### **1.2 Recursos materiales para el desarrollo de la solución propuesta**

Los recursos materiales resultan ser primordiales y necesarios si una organización dada quiere llevar a cabo o concretar una acción de manera exitosa. Sin ellos seguramente fracasará o por lo pronto no alcanzará los objetivos que se proponga en un ciento por ciento (16). A continuación se muestran en la **Tabla 1** el desglose de los recursos materiales estudiados y los gastos asociados a los mismos.

**Tabla 1: Desglose de los recursos materiales y sus gastos asociados.**

<b>Recursos materiales</b>	<b>Ejemplos</b>	<b>Gastos</b>
Inmuebles	<ul style="list-style-type: none"> <li>• Oficinas</li> <li>• Talleres</li> <li>• Terrenos</li> </ul>	Depreciación (oficinas y talleres)
Elementos de oficina	<ul style="list-style-type: none"> <li>• Mobiliarios (escritorio, sillas, estantes, libros)</li> <li>• Calculadora</li> <li>• Paquetes de hojas</li> <li>• Bolígrafos</li> <li>• Carpetas</li> </ul>	Depreciación de mobiliarios Gasto de materiales de oficina
Equipos electrónicos	<ul style="list-style-type: none"> <li>• Ordenadores</li> <li>• Impresoras</li> <li>• Fotocopiadoras</li> <li>• Fax</li> <li>• Video proyector</li> </ul>	Depreciación de equipos electrónicos
Instalaciones	<ul style="list-style-type: none"> <li>• Edificios</li> <li>• Instrumentos</li> <li>• Herramientas</li> <li>• Maquinarias</li> </ul>	Depreciación de instalaciones
Medios de transporte	<ul style="list-style-type: none"> <li>• Vehículos (aéreos, terrestres, marítimos)</li> </ul>	Gasto de combustible Gasto de reparaciones Depreciación de vehículos
Medios de comunicación	<ul style="list-style-type: none"> <li>• Teléfonos</li> <li>• Fax</li> <li>• Internet (correo, redes sociales)</li> <li>• Transporte</li> </ul>	Gasto de medios de comunicación
RRHH	<ul style="list-style-type: none"> <li>• Personas</li> </ul>	Gasto de hospedaje Gasto viáticos Gasto de personal (salarios, aporte a la seguridad social impuesto sobre la fuerza de trabajo, vacaciones)
Materiales de empaque	<ul style="list-style-type: none"> <li>• Cajas</li> <li>• Bolsas</li> <li>• Precintas</li> </ul>	
Materiales auxiliares	<ul style="list-style-type: none"> <li>• Agua</li> <li>• Materias primas</li> </ul>	Gasto de materiales de empaque

## 1.3 Estado del arte

### 1.3.1 Sistemas de planificación existentes a nivel mundial

#### 1.3.1.1 OpenProj

OpenProj fue desarrollado en el 2007 por la compañía Projity, es un *software* multiplataforma implementado en Java, funciona en varios sistemas operativos como es el caso de Windows, Linux, Unix y Mac. Permite la planificación y ejecución oportuna y organizada de nuestro proyecto de *software* (17). Es una aplicación gratuita, de código abierto para la gestión de proyectos, donde se pueden crear cuantos proyectos sean convenientes. Soporta un ilimitado número de tareas en una línea de tiempo donde se asignan prioridades a cada una de ellas y permite organizarlas de diferentes formas y estilos. Controla todos los aspectos referentes a la gestión de proyectos como la planificación, programación, la gestión y asignación de recursos. Permite trabajar con proyectos de gran envergadura y con calendarios, en los que se incluirán las actividades (18). Proporciona las siguientes funcionalidades para la gestión de planificaciones:

- Gestión de calendarios de trabajo.
- Existencia de diferentes vistas del proyecto (diagramas de Gantt, diagramas de red, histogramas).
- Gestión de tareas, con niveles de jerarquía y todas las posibilidades de dependencia necesarias.
- Gestión de recursos (tantos humanos como materiales) asignados a un proyecto.
- Funciones de seguimiento de proyectos.

OpenProj permite la asignación de los recursos a cada una de las tareas y a partir de ello se determina el costo de ejecución, se tiene en cuenta además si incluye subtareas y a su vez los recursos que hayan sido asignadas a estas (19).

#### 1.3.1.2 Project

Project, es un *software* de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Es un *software* privativo, funciona en el sistema operativo Windows. Proporciona herramientas que ayudan en el proceso de planificación y ejecución de un proyecto, donde se destacan: el Diagrama de Gantt, que se encarga del tiempo en el que se programan las tareas y el Diagrama Pert, que aclara la relación entre las mismas. Entre las ventajas que ofrece este *software* se encuentran:

- Crear un plan realista del proyecto: gracias a los asistentes de planificación, una vez que se han establecido las tareas a desarrollar y los recursos con los que se cuenta, se puede aumentar la efectividad del proyecto basándonos en sugerencias y trucos que estos ofrecen.
- Gestionar el proyecto y ajustarse a los cambios: esto lleva consigo el seguimiento de las tareas, si se van realizando como se esperaban, si hay algún retraso en su finalización, además se tiene en cuenta los imprevistos que puede haber.
- Comunicación de los resultados del proyecto: como generalmente son muchas las personas que participan en un proyecto, éstas tienen que estar informadas para que puedan trabajar efectivamente. Mediante el uso de los informes, se puede presentar la información necesaria en cada caso.
- Evaluación del proyecto: Project almacena toda la información relacionada con las tareas y la administración del presupuesto, esta información ayuda a realizar la evaluación final del proyecto, además permite hacer sugerencias para proyectos futuros.

Project permite generar un informe del presupuesto del proyecto de *software*, donde enumera todas las tareas del presupuesto ordenadas por el costo total. El informe refleja los costos reales, el costo previsto (programado), la variación, el costo fijo, el método de acumulación y el costo restante. Permite enumerar las tareas de mayor a menor costo, así como asignarle recursos a las mismas si estos recursos están disponibles para ser utilizados (21).

### 1.3.1.3 GESPRO

GESPRO es un ecosistema para el desarrollo y la innovación en gestión de proyectos, desarrollado por el laboratorio de investigaciones en gestión de proyectos de la Universidad de las Ciencias Informáticas de Cuba en el año 2008. Es un *software* multiplataforma implementado bajo un ecosistema basado en tecnologías libres. Esta herramienta permite el almacenamiento de los datos y ayuda a la toma de decisiones, la planificación y control de los proyectos de *software*. Es un *software* genérico y adaptable, para cualquier tipo de organización que desee contratar sus servicios. Combina el uso de una solución informática para la gestión de proyectos y un sistema de formación especializada en los proyectos de *software*. Posibilita no sólo la informatización de la gestión de proyectos en las organizaciones, sino también la mejora continua de sus procesos de planificación, seguimiento y control. Entre las principales funcionalidades de la solución informática se encuentran:

- Gestión de tiempo: permite la definición de las actividades, la secuencia de las actividades y la planificación colaborativa de proyectos.

- Gestión de costos: permite la planificación de los recursos, estimación de los costos, gestión de indicadores para la gestión de los costos y finanzas.
- Gestión de recursos humanos: permite la gestión por competencias, procesos requeridos para realizar un uso eficiente y eficaz de los recursos humanos, además del análisis del aprovechamiento de las fuerzas productivas, la eficacia y la eficiencia en el desarrollo de proyectos.
- Gestión logística: permite la gestión del fondo de recursos compartidos, gestión de adquisiciones hasta la asignación a proyectos y gestión logística además de los contratos con proveedores.
- Gestión financiera: permite gestionar los ingresos y gastos del proyecto.
- Generación de tareas en los cronogramas de los proyectos a partir de los procesos de la organización: garantiza de esta forma la estandarización en las actividades de los procesos de la organización.

GESPRO permite la planificación del alcance y el tiempo, la gestión de recursos humanos y sus competencias, así como la planificación de los recursos al proyecto de *software* de manera general, estima los costos y gestiona los indicadores para evaluar los mismos (20).

### 1.3.2 Resolución del Ministerio de Finanzas y Precios

Según Lina O. Pedraza Rodríguez, los clasificadores por objeto del gasto son el ordenamiento sistemático y homogéneo de los gastos y las variaciones de activos y pasivos que el sector gobierno aplica en el desarrollo de sus actividades. Responde en esencia a la pregunta en que se gasta y constituye un instrumento informativo para el análisis y seguimiento de la gestión financiera de las finanzas públicas.

A continuación se presentan los clasificadores por objeto del gasto del presupuesto de estado, extraídos de la resolución no.354/2013 de la ministra de finanzas y precios (Lina O. Pedraza Rodríguez).

#### **Partida 11: Materias primas y materiales**

Alimento: gastos corrientes asociados al insumo de productos alimenticios entregados para la oferta del servicio de alimentación. Incluyen entre otros: bebidas, productos agrícolas, merienda escolar, también incluye alimentos de consumo animal para aquellas unidades presupuestadas autorizadas al insumo de productos alimenticios y agropecuarios con destino a la alimentación de animales de corral, de trabajo o exhibición.

Materiales para la enseñanza: gastos corrientes asociados al insumo de los materiales y utensilios utilizados para impartir la enseñanza en cualquiera de los niveles educacionales: libros, libretas, cuadernos, folletos, lápices, gomas, bolígrafos, reglas, tizas, mapas, atlas, pizarras, pantallas, borradores y otros afines.

Materiales y artículos de consumo: gastos corrientes asociados al insumo de los siguientes materiales: papel de escritorio, papel de computación, materias primas y materiales, pintura, artículos de limpieza y aseo, materiales eléctricos, interruptores, lámparas y otros.

Útiles y herramientas: gastos corrientes asociados al insumo de materiales y artículos de oficina que incluye entre otros: útiles de escritorio y oficina, útiles y herramientas, utensilios de cocina y utensilios deportivos.

Partes y piezas de repuestos: gastos corrientes asociados al insumo de piezas de repuestos y accesorios necesarios para el funcionamiento y mantenimiento del parque automotor y otros equipos.

### **Partida 30: Combustibles y lubricantes**

Gas: gastos corrientes asociados al consumo de gas manufacturado y gas licuado.

Combustibles: gastos corrientes asociados al consumo de gasolina, diésel y otros tipos de combustibles.

### **Partida 50: Gastos de personal**

Salario: gastos asociados a las remuneraciones, que de acuerdo con la legislación laboral y salarial vigente, recibe el trabajador contratado por tiempo determinado e indeterminado por concepto de salario, de acuerdo con los sistemas de pagos aprobados para cada entidad, incluyendo además los siguientes: otras retribuciones salariales a los trabajadores, remuneraciones a los graduados en el período de adiestramiento, licencias deportivas, estipendios y estimulación al personal.

Acumulación de vacaciones: gastos asociados a la acumulación mensual para el pago de vacaciones a los trabajadores contratados por tiempo determinado e indeterminado, de acuerdo con la legislación laboral vigente.

### **Partida 70: Depreciación y amortización**

Depreciación activos fijos tangibles: gastos en que se incurren por la depreciación de sus activos fijos tangibles, de acuerdo a la legislación financiera vigente.

### **Partida 80: Otros gastos monetarios**

Viáticos: gastos corrientes en que incurre su personal cuando en funciones de trabajo viaja fuera de la localidad de residencia y al exterior del país, de acuerdo a lo que en materia de dietas establece la legislación financiera vigente que incluye entre otros los siguientes gastos: alimentación, transportación, alojamiento y gastos de bolsillo.

Servicios de mantenimiento y reparaciones corrientes: gastos asociados a la contratación de mantenimientos y reparaciones de muebles y equipos que incluye entre otros: mantenimiento de vehículos, maquinarias y equipos, equipos de computación y otros.

Otros servicios contratados: gastos corrientes asociados a la contratación de servicios tales como agua, teléfono, fax, servicios gastronómicos, otros servicios de telecomunicaciones, transportación, entre otros.

### **Partida 82: Presupuesto de la seguridad social**

Pensiones a corto plazo: gastos destinados a cubrir temporalmente actividades de la seguridad social. Incluye los subsidios pagados a trabajadores enfermos de tuberculosis pulmonar, de acuerdo a la legislación vigente, la seguridad social a corto plazo del sector privado, las prestaciones económicas a la madre trabajadora según lo establecido en la legislación vigente, las pensiones por invalidez parcial, así como otros conceptos de gastos que se determinen de acuerdo a las legislaciones vigentes y que se les reintegra a los empleadores.

Pensiones a largo plazo: gastos destinados a cubrir las pensiones y rentas vitalicias pagadas por la seguridad social.

### **1.4 Valoración de los sistemas de gestión estudiados**

Después de realizar el estudio de las soluciones informáticas antes presentadas; donde se tuvieron en cuenta no sólo los aspectos fundamentales desde el punto de vista del *software*, sino también del producto, se evidencia la no existencia de un sistema de planificación que incluya el proceso de evaluación de los gastos para las actividades de aseguramiento interno, que cumpla además con lo establecido en la Instrucción no.1 y la Resolución del Ministerio Finanzas y Precios. Independientemente de lo antes mencionado, servirán de base para la identificación de una serie de funcionalidades a incluir en la solución propuesta. A continuación se muestra en la **Tabla 2** la comparación de diferentes indicadores asociados al estudio de los sistemas.

Tabla 2: Comparación de los sistemas.

Indicadores a evaluar	Sistemas		
	Project	OpenProj	GESPRO
Asignación de recursos a las actividades	SI	SI	NO
Integración con otros sistemas	NO	SI	SI
Software libre	SI	NO	NO

### 1.5 Metodología de desarrollo

Una metodología de desarrollo es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto *software* desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado (37). La UCI define para su actividad productiva hacer una variación de la metodología *Agile Unified Process* (AUP) la cual aplica técnicas de desarrollo ágiles. AUP-UCI cuenta con 3 fases (Inicio, Ejecución, Cierre) y en la fase de Ejecución es donde se centra el desarrollo del presente trabajo.

A continuación se detallan las disciplinas y los artefactos a generar por cada etapa del ciclo de vida de esta metodología de desarrollo teniendo en cuenta su adaptación para el desarrollo del presente trabajo.

**Modelado de negocio:** en esta disciplina se realiza una comprensión de los procesos de negocio para entender su funcionamiento y analizar si la solución propuesta va a cumplir su propósito. Se obtendrá el diagrama de proceso de negocio, modelo conceptual, glosario de términos y reglas de negocio.

**Requisitos:** en esta disciplina el objetivo principal es proveer a los desarrolladores del sistema un mejor entendimiento de los requerimientos los cuales deben ser completos y consistentes. Por completos se entiende que todos los servicios requeridos por el usuario deben especificarse y la consistencia significa que ninguna definición de requisitos debe contradecir otra. Se obtendrá la especificación y descripción de requisitos de *software*.

**Análisis y diseño:** en esta disciplina se modela la solución propuesta. Se obtendrá el diagrama de clases del diseño, diagrama de secuencia, diagrama de componentes, modelo de datos y diseño de casos de prueba.

**Implementación:** a partir de los resultados del análisis y diseño se implementa el proceso de funcionamiento interno. Se obtendrá el código fuente de la solución propuesta.

**Pruebas interna:** en esta disciplina se verifica el resultado de la implementación identificándose posibles errores en la documentación y el *software*. Se obtendrán las no conformidades.

## 1.6 Patrones de diseño

Un patrón de diseño es una estructura que resuelve un problema de diseño particular dentro de un contexto específico y en medio de “fuerzas” que pueden tener un impacto en la manera en que se aplica y utiliza el patrón (22). A continuación se presentan los patrones de diseño a utilizar en la evaluación de los gastos para las actividades de aseguramiento interno.

### 1.6.1 Patrones GRASP

GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de *software* para asignar responsabilidades). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (23).

Los principales patrones GRASP son: experto, creador, alta cohesión, bajo acoplamiento, controlador.

**Experto:** su función esencial se basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen (23).

**Creador:** su función esencial se basa en identificar quién debe ser el responsable de la creación de nuevos objetos. Asignarle a la clase B la responsabilidad de crear una instancia de la clase A. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento (23).

**Alta cohesión:** la principal función de este patrón es asignar una responsabilidad de modo que la cohesión siga siendo alta. La alta cohesión es un principio que debemos tener presente en todas las

decisiones de diseño: es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño (23).

**Bajo acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Por lo que su función esencial es asignar una responsabilidad de modo que el acoplamiento siga siendo bajo. Acoplamiento bajo significa que una clase no depende de muchas clases. Beneficia la reutilización y permite que no se afecten los cambios en otros componentes (24).

**Controlador:** su función esencial se basa en controlar el flujo de eventos del sistema. Es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación. El controlador no realiza mucho trabajo por sí mismo; más bien coordina la actividad de otros objetos. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control (24).

### 1.6.2 Patrones GOF

Los patrones GOF (*Gang of Four* o Banda de los cuatro) describen soluciones simples y elegantes a problemas específicos en el diseño de *software* orientado a objetos y otros ámbitos referentes al diseño de interacción o interfaces (2).

Los patrones GOF recogen 23 patrones de diseño comunes que se clasifican en:

1. Creacionales
2. Estructurales
3. De comportamiento

**Fachada:** patrón estructural que permite proveer una interfaz unificada y sencilla como intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas (25).

**Cadena de responsabilidad:** patrón de comportamiento que se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición (25).

## 1.7 Lenguajes, tecnologías y herramientas propuestas para el desarrollo de la solución

A continuación se presentan los lenguajes, tecnologías y herramientas utilizadas para el desarrollo de la solución propuesta. Estas fueron seleccionadas por ser las definidas por el Departamento de aplicaciones de gestión empresarial del centro CEIGE para el desarrollo de soluciones informáticas.

### 1.7.1 Lenguajes de modelado

#### **UML**

UML (*Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de *software* orientado a objetos, entre sus principales ventajas se puede usar para modelar distintos tipos de sistemas: sistemas de *software*, sistemas de *hardware*, y organizaciones del mundo real. UML es una consolidación de muchas de las notaciones y conceptos más usadas orientadas a objetos, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de *software* reutilizables (26).

#### **BPMN**

BPMN (*Business Process Modeling Notation*) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo, su principal objetivo es proveer una notación estándar que sea fácilmente legible y entendible y servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación (27).

### 1.7.2 Lenguajes de desarrollo del lado del cliente

#### **HTML 5**

HTML (*Hypertext Markup Language* o Lenguaje de Marca de Hipertexto) es un lenguaje de marca independiente de la plataforma y de la máquina, que utiliza marcas (*tags* o etiquetas) para describir los documentos, además de vínculos de hipertexto para conectar y desplazarse de un documento a otro. Se integra con lenguajes como JavaScript y PHP y no todos los navegadores interpretan este lenguaje de la misma forma, lo cual puede ser visto como una de sus desventajas (28).

#### **CSS 3**

Las hojas de estilo en cascada (*Cascading Style Sheets*) permiten aplicar, mediante una sintaxis especial, los estilos de presentación de páginas web en un formato mucho más exacto. Este lenguaje ayuda a mejorar y definir el aspecto visual de la página en distintos navegadores, mientras que separa la parte semántica (HTML) de la presentacional (*style sheets*) (29).

#### **JavaScript**

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que

Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web (30).

### 1.7.3 Lenguajes de desarrollo del lado del servidor

#### PHP 5.2

PHP permite el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código abierto, es el más popular y extendido en la web. Es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Por otra parte con respecto a otros lenguajes de programación como Java o C++ es mucho menos robusto y la programación orientada a objetos es aún muy deficiente para aplicaciones grandes (3).

### 1.7.4 Tecnologías y herramientas de desarrollo

#### Sauxe 1.0

Sauxe como cualquier marco de trabajo en el desarrollo de *software*, brinda una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Este marco de trabajo, fusionado bajo tecnología totalmente libre (entre ellas PHP, Postgresql, Apache) posee el desarrollo de tecnologías propias basadas en otros marcos de trabajo como ZendFramework para el manejo de la lógica de negocio, Doctrine para el acceso a datos y ExtJS para la capa de presentación. Cuenta con una arquitectura en capas y contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. Siguiendo el paradigma de independencia tecnológica por el cual apuesta el país (31).

#### ZendFramework 1.7

Para el manejo de la lógica del negocio se utilizará el marco de trabajo ZendFramework. Es un marco de trabajo de código abierto, orientado a objeto; está diseñado para la versión 5 de PHP. Posee muchos componentes acoplados que se pueden utilizar o no independientes. Establece una estructura básica utilizando los componentes del patrón Modelo-Vista-Controlador (MVC). Posee

componentes con estructuras para AJAX, para facilitar peticiones y dar respuestas basadas en AJAX y otros componentes para facilitar la composición de vistas (32).

### **Doctrine 1.2.1**

Doctrine se utilizará para trabajar con un esquema de base de datos como si fuese un conjunto de objetos y no de tablas y registros. Está inspirado en Hibernate, que es uno de los ORM (*Object Relational Mapper*) más populares y grandes que existen; brinda una capa de abstracción de la base de datos muy completa, sin necesidad de duplicar códigos, facilitando un mejor trabajo a los diseñadores SQL (32).

### **ExtJS 2.2**

ExtJS es una librería JavaScript que permite construir aplicaciones complejas en internet, además de flexibilizar el manejo de componentes de la página como el DOM, comunicación con el servidor usando AJAX, permite crear interfaces de usuario bastante funcionales que es principalmente para lo cual se emplea, posee numerosas funcionalidades que permiten añadir interactividad a las páginas HTML. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos. Se distribuye la carga de procesamiento Cliente- Servidor, permitiendo que el servidor pueda atender más clientes al mismo tiempo (32).

### **AJAX**

Ajax por sus siglas en inglés *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML) es la técnica de desarrollo web que permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Surge de la combinación de las tecnologías ya existentes:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías (33).

### **Sistema de control de versiones: subversión 1.6**

Subversión (SVN) es la herramienta de entorno colaborativo que se utiliza para el control de versiones. Se encuentra preparado para funcionar en red y se distribuye bajo licencia libre. Mantiene versiones no sólo de archivos, sino también de directorios y versiones de los metadatos asociados a

esos directorios. Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre. Brinda facilidades de soporte tanto de ficheros de texto como binarios (2).

**Herramienta CASE: *visual paradigm for UML 8.0 enterprise edition***

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Constituye una herramienta de *software* libre de gran utilidad para el analista (34).

**Entorno integrado de desarrollo (IDE): netbeans 8.0**

NetBeans es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris, de código abierto escrito completamente en Java y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones del tipo web y empresarial, aplicaciones de escritorio y aplicaciones móviles, utilizando la plataforma Java, así como un número importante de módulos para extenderlo a otros lenguajes (Oracle) (34).

**Servidor web: apache 2.2**

Se utilizará como servidor web Apache pues es una tecnología gratuita de código abierto compatible con muchos Sistemas Operativos. Tiene todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configurabilidad en la creación y gestión de registros de actividad. Apache permite la creación de ficheros de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (31).

**Sistema gestor de bases de datos: postgresql 9.1**

Es un sistema de gestión de bases de datos, basado en el proyecto Postgres de la Universidad de Berkeley. Está considerado como el sistema de base de datos de código abierto más avanzado del mundo, publicado bajo la licencia BSD. Soporta distintos tipos de datos: soporte para los tipos base, soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios. Como toda herramienta de *software* libre PostgreSQL tiene entre otras ventajas las de contar con una gran comunidad de desarrollo en

Internet, su código fuente está disponible sin costo alguno y algo muy importante es que es multiplataforma (34).

### **Navegador web: mozilla firefox 33**

Mozilla Firefox, o simplemente Firefox es un navegador web libre y de código abierto desarrollado por Mozilla, una comunidad global que trabaja junta para mantener una Web Abierta, pública y accesible para todos. Cuenta con alucinantes características entre las que se encuentran: navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas, navegación privada, y muchas otras más (35). Tiene gran compatibilidad con los estándares web (señálese HTML, XML, SS y JavaScript). Este importante navegador funciona con Windows, Linux, Mac OS X y varios sistemas operativos más (36).

### **Conclusiones parciales**

Con el desarrollo del presente capítulo se arribó a las siguientes conclusiones:

- El análisis de diferentes sistemas informáticos existentes relacionados con el objeto de estudio, evidenció que no existe un sistema de planificación que integre la información referente a la evaluación de los gastos de las actividades de funcionamiento interno que cumpla con lo establecido a nivel nacional.
- El estudio de la resolución no.354/2013 del Ministerio de Finanzas y Precios, permitió identificar los principales recursos materiales a utilizar en el proceso de evaluación de gastos para las actividades de aseguramiento interno.
- La identificación de los lenguajes, tecnologías y herramientas, así como el estudio realizado sobre el modelo de desarrollo a utilizar en la propuesta de solución, aportó conocimiento sobre los mismos, lo que facilitará su posterior utilización.

## **CAPÍTULO II: ANÁLISIS Y DISEÑO**

### **Introducción**

El siguiente capítulo muestra una descripción de los elementos esenciales a tener en cuenta para el desarrollo de la solución propuesta. Recoge los artefactos generados con la utilización de la metodología de desarrollo utilizada como parte del proceso de desarrollo para dar solución al problema planteado. Se describen las características que debe tener la evaluación de los gastos de funcionamiento interno a partir de la especificación y descripción de los requisitos funcionales y no funcionales que debe tener el sistema además se definen los patrones de diseño utilizados.

### **2.1 Modelado del negocio**

#### **2.1.1 Diagrama de procesos de negocio**

Los procesos de negocio constituyen una secuencia lógica y cronológica de las acciones que se deben realizar, cada vez que se produce el suceso que lo origina en una organización. En otros términos un proceso de negocio es una serie de actividades que se desarrollan y que tienen como objetivo en su conjunto, producir un mayor valor para el cliente que lo recibe (38). A continuación se presenta la actualización del proceso de negocio Establecer plan del SIPAC, en el cual se identifica la definición del presupuesto del plan para actividades de aseguramiento interno, como una actividad que se ejecuta como parte de dicho proceso de negocio, donde se resalta en color negro.

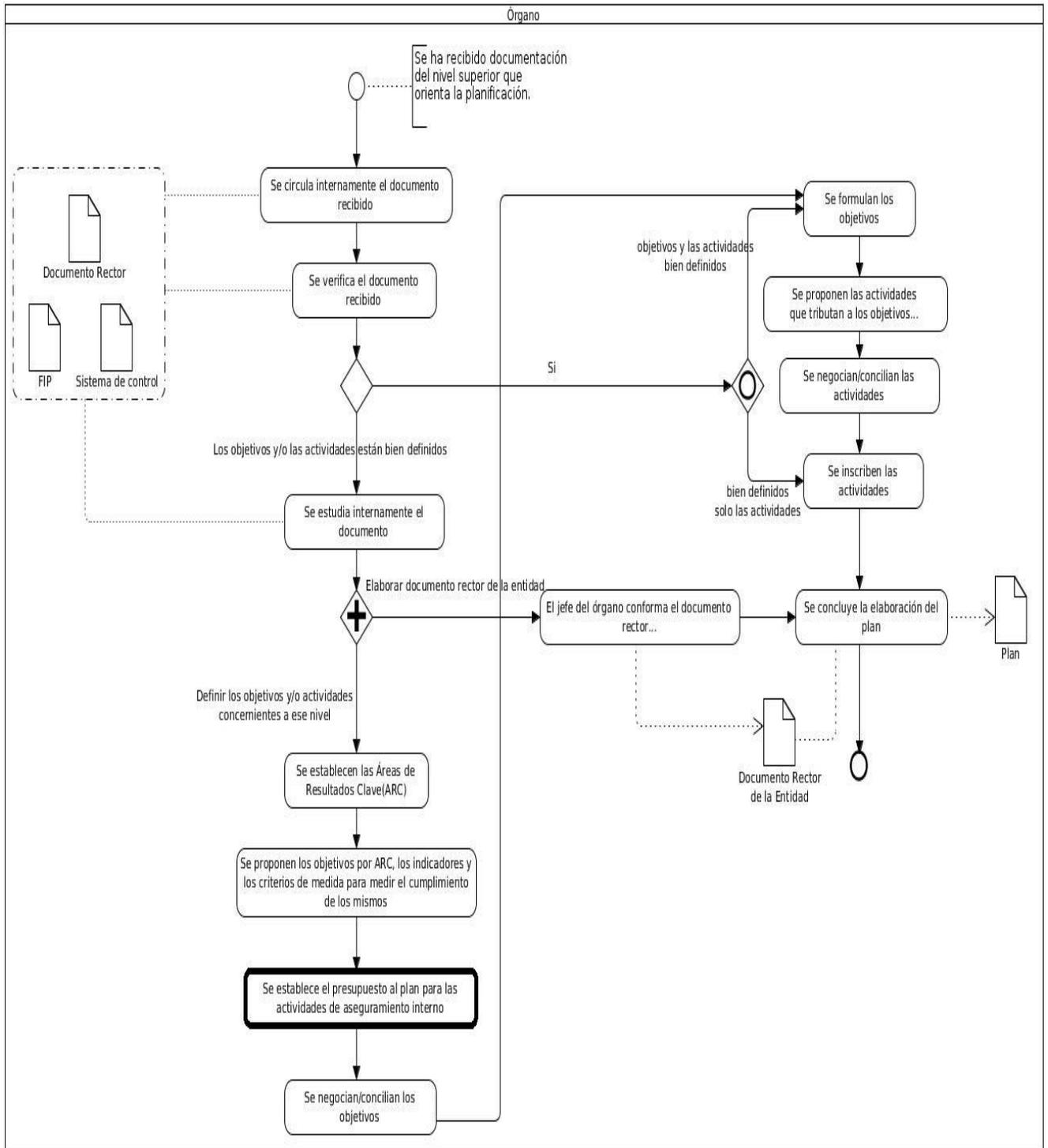


Figura 1: Diagrama de proceso de negocio Establecer plan.

2.1.2 Especificación del proceso de negocio: Establecer plan

**Tabla 3: Especificación de proceso de negocio: Establecer plan.**

<b>Objetivo</b>	Prever con anticipación los objetivos y actividades que se realizarán para alcanzarlos.
<b>Evento(s) que lo genera(n)</b>	Se ha recibido documentación del nivel superior que orienta la planificación.
<b>Pre condiciones</b>	Se ha emitido una documentación que orienta la planificación en el nivel superior.
<b>Marco legal</b>	N/A.
<b>Clientes internos</b>	Puntualizar plan. Establecer sistema de control. Aprobar plan.
<b>Clientes externos</b>	N/A.
<b>Entradas</b>	Documento rector (Documento word). Sistema de control (Documento word). FIP <sup>1</sup> (Documento word).
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
<ol style="list-style-type: none"> <li>1. Se circula internamente el documento. El documento en formato impreso o digital se circula a todos los involucrados del mismo nivel de dirección que participarán en el establecimiento del plan.</li> <li>2. Se verifica el documento recibido. Este documento se identifica para conocer si es un documento rector, un FIP o un sistema de control. En caso de ser un sistema de control o un FIP los objetivos y/o actividades estarán bien definidos. En el caso de que se trate de un documento rector se procederá a estudiarse para a partir de entonces, definir los objetivos y/o actividades concernientes a este nivel en la actividad 3 del flujo básico: Se estudia internamente el documento. Para el primer caso, si el documento recibido tiene bien definidos los objetivos y las actividades se ejecuta el paso 2 del flujo paralelo Definir plan, pero si por el contrario el documento recibido tiene bien definidos solo las actividades entonces se ejecuta el paso 7 del flujo paralelo Definir plan.</li> <li>3. Se estudia internamente el documento. El documento es estudiado minuciosamente por parte de todos los involucrados del nivel de dirección que se trate para tener todos los elementos necesarios que ayuden a establecer el plan a este nivel.</li> <li>4. Se ejecutan los flujos paralelos <u>Definir plan y Elaborar documento rector de la entidad.</u></li> </ol>	
<b>Pos-condiciones</b>	
1. N/A.	
<b>Salidas</b>	

<sup>1</sup> Factores que Influyen en el Plan (FIP): es un documento que contiene elementos que desencadenan cambios en los planes, el mismo es emitido por el nivel superior o propio.

1. N/A.

### **Flujos paralelos**

#### **Flujos paralelo 5 Definir plan**

1. Se establecen las áreas de resultados clave. Si es necesario se crean nuevas áreas de resultados clave ya sea porque se asumen las indicadas por el nivel superior (todas o parte de ellas) o se generan las propias a este nivel.
2. Se proponen los objetivos por área de resultados clave, los indicadores y los criterios de medida para medir el cumplimiento de los mismos. Cada implicado en el proceso de elaboración de los objetivos hace su propuesta de objetivos, los indicadores y los criterios de medida enmarcados en las correspondientes áreas de resultados clave.
3. Se establece el presupuesto al plan para las actividades de aseguramiento interno. Al crear el plan se establece el presupuesto planificado para ejecutar las actividades de aseguramiento interno del plan en cuestión.
4. Se negocian/concilian los objetivos. Los objetivos se llevan a un proceso de concertación entre los implicados para decidir por consenso los objetivos a incluir en el plan.
5. Se formulan los objetivos. Los objetivos conciliados se formulan (se redactan correctamente) y se inscriben en el plan.
6. Se proponen las actividades que tributan a los objetivos correspondientes. Cada implicado hace su propuesta de actividades que tributen a los objetivos previamente formulados.
7. Se negocian/concilian las actividades. Las actividades que han sido propuestas por los implicados se llevan a un proceso de concertación entre ellos para definir las que finalmente serán incluidas en el plan.
8. Se inscriben las actividades. Las actividades que han sido conciliadas se inscriben en el plan. Estas tributarán a sus correspondientes objetivos como se definió durante la actividad de conciliación de las actividades.
9. Se concluye la elaboración del plan. Se conforma el documento final de plan para posteriormente ser sometido a aprobación por el responsable de aprobar. El jefe del órgano de dirección a este nivel conforma el documento rector para ser entregado a los subordinados del nivel inferior.
10. Concluye el proceso.

### **Pos-condiciones**

1. Se ha establecido el plan.

### **Salidas**

1. Plan (Documento word).

#### **Flujos paralelo 5 Elaborar documento rector de la entidad.**

1. El jefe del órgano conforma el documento rector para el nivel inmediato inferior. A medida que se van definiendo las áreas de resultados clave, los objetivos y las actividades el jefe del órgano elabora un documento rector propio de la entidad que será entregado al nivel inmediato inferior para que este a su vez defina su documento rector y su plan.
2. Continúa en el paso 8 del flujo alterno Definir plan.

### **Pos-condiciones**

1. Se ha elaborado documento rector de la entidad.
<b>Salidas</b>
1. Documento rector de la entidad (Documento word).
<b>Flujos alternos</b>
1. N/A.
<b>Pos-condiciones</b>
1. N/A.
<b>Salidas</b>
1. N/A.
<b>Asuntos pendientes</b>
1. N/A.

### 2.1.3 Modelo conceptual

El modelo conceptual es una representación de los conceptos fundamentales y sus relaciones de la situación que se analiza para una mejor comprensión de la misma. A continuación se describen los principales conceptos relacionados a la solución propuesta, los cuales se encuentran representados en la **Figura 2** a través del modelo conceptual y fueron extraídos del expediente de proyecto SIPAC 3.0 del artefacto glosario de términos.

**Elemento de la planificación:** agrupa a los elementos asociados a la planificación, estos pueden ser: planes, objetivos y actividades.

**Actividades:** conjunto de operaciones o tareas, propias de una persona o entidad, destinadas para cumplir determinado(s) objetivo(s).

**Área de Resultado Clave (ARC):** término empleado para definir un resultado cuyo logro implica la participación de dos o más elementos (departamento, dirección, grupo de trabajo, etc.) de la estructura funcional de una entidad. Tal logro es esencial para que la organización cumpla su misión.

**Objetivos:** metas a alcanzar que se traza cada entidad.

**Plan:** es un modelo sistemático que se elabora antes de realizar una acción, con el propósito de dirigirla y encauzarla. En este sentido, un plan también es un documento que precisa los detalles necesarios para realizar una misión.

La **Figura 2** representa el modelo conceptual asociado al proceso de evaluación de los gastos de funcionamiento interno para el Sistema de Planificación de Actividades SIPAC.

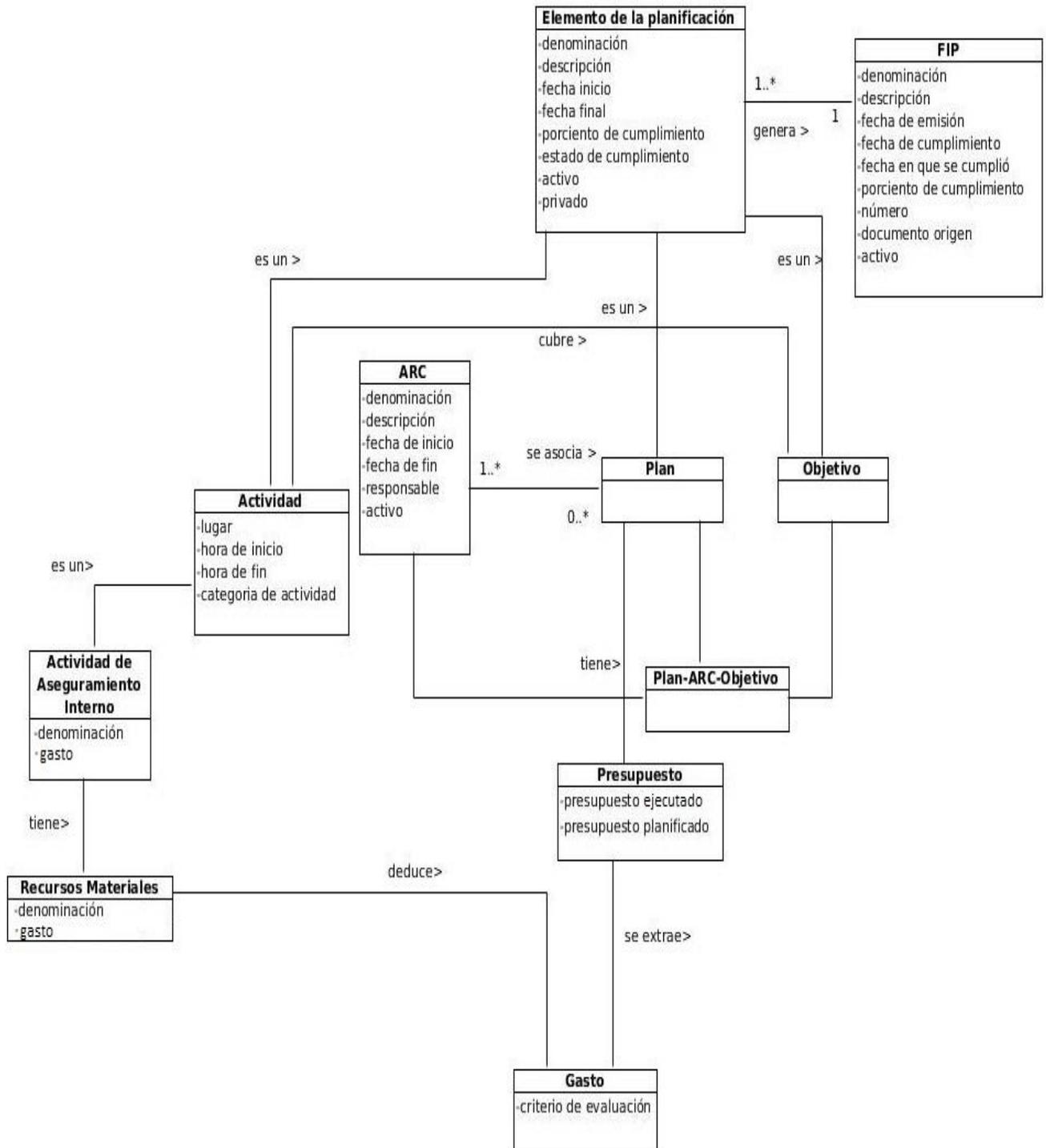


Figura 2: Modelo conceptual de los gastos de funcionamiento interno para SIPAC.

## 2.2 Requisitos

Un requisito de *software* es, según la IEEE (*Institute of Electrical and Electronics Engineers*, en español Instituto de Ingenieros Eléctricos y Electrónicos) una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Se puede ver también como una condición o capacidad que tiene que ser alcanzada o poseída por un sistema para satisfacer formalmente un contrato definido entre las partes interesadas (2).

### 2.2.1 Técnicas de captura de requisitos

Para obtener los requisitos funcionales de la solución propuesta se utilizaron las siguientes técnicas de captura de requisitos:

- **Talleres:** con los miembros del proyecto SIPAC, dirigido por la jefa de proyecto, se intercambiaron ideas y opiniones identificando y detallando los requisitos necesarios para cumplir con las expectativas del cliente.
- **Entrevistas:** fue entrevistado el cliente además de profesionales del proyecto SIPAC, con amplios conocimientos sobre el objeto de estudio, lo que permitió obtener toda la información posible sobre la visión de los requisitos del sistema y comprender los propósitos de la solución buscada.

### 2.2.2 Requisitos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas pueden declarar explícitamente lo que el sistema no debe hacer (39).

Para la evaluación de los gastos de funcionamiento interno fueron identificados los siguientes requisitos funcionales (RF).

#### **RF1:** Gestionar gastos de elementos de oficina

1.1 Registrar gastos de elementos de oficina

1.2 Modificar gastos de elementos de oficina

1.3 Eliminar gastos de elementos de oficina

#### **RF2:** Gestionar gastos de depreciación de equipos electrónicos

2.1 Registrar gastos de depreciación de equipos electrónicos

2.2 Modificar gastos de depreciación de equipos electrónicos

2.3 Eliminar gastos de depreciación de equipos electrónicos

**RF3:** Gestionar gastos de medios de transporte

3.1 Registrar gastos de medios de transporte

3.2 Modificar gastos de medios de transporte

3.3 Eliminar gastos de medios de transporte

**RF4:** Gestionar gastos de depreciación de vehículos

4.1 Registrar gastos de depreciación de vehículos

4.2 Modificar gastos de depreciación de vehículos

4.3 Eliminar gastos de depreciación de vehículos

**RF5:** Gestionar gastos de medios de comunicación

5.1 Registrar gastos de medios de comunicación

5.2 Modificar gastos de medios de comunicación

5.3 Eliminar gastos de medios de comunicación

**RF6:** Gestionar gastos de salario básico

6.1 Registrar gastos de salario básico

6.2 Modificar gastos de salario básico

6.3 Eliminar gastos de salario básico

**RF7:** Gestionar gastos de RRHH

7.1 Registrar gastos de RRHH

7.2 Modificar gastos de RRHH

7.3 Eliminar gastos de RRHH

**RF8:** Gestionar gastos de materiales de empaque

8.1 Registrar gastos de materiales de empaque

8.2 Modificar gastos de materiales de empaque

8.3 Eliminar gastos de materiales de empaque

**RF9:** Gestionar presupuesto al plan

9.1 Definir presupuesto al plan

9.2 Listar presupuesto al plan

9.3 Analizar presupuesto

9.4 Calcular presupuesto ejecutado según actividades de aseguramiento interno

**RF10:** Gestionar recursos materiales a las actividades

10.1 Listar gastos de los recursos materiales a las actividades

10.2 Eliminar recursos materiales a las actividades

10.3 Modificar recursos materiales a las actividades

10.4 Buscar recursos materiales

10.5 Calcular gasto a las actividades

**2.2.3** Especificación de requisitos funcionales

A continuación se muestra la especificación del requisito **Registrar gastos de elementos de oficina**, la especificación de los restantes requisitos funcionales se encuentran descritos en el expediente de proyecto SIPAC 3.0.

**Tabla 4: Especificación del requisito: Registrar gastos de elementos de oficina.**

<b>Precondiciones</b>	El usuario ha accedido a la interfaz Adicionar actividad. El usuario ha seleccionado la opción Recursos.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar gastos de elementos de oficina</b>	
1.	El usuario selecciona el elemento de oficina que desea registrarle el gasto dando clic sobre el mismo.
2.	El sistema muestra una interfaz que permite al usuario introducir los siguientes datos del elemento de oficina seleccionado: <ul style="list-style-type: none"> <li>• Cantidad de unidades</li> <li>• Precio de la unidad</li> </ul>
3.	El usuario introduce los datos del elemento de oficina y da clic en el botón Registrar.
4.	El sistema valida que los datos introducidos estén correctos y los registra.
5.	Concluye el requisito.
<b>Pos-condiciones</b>	
1.	Se registra en el sistema el gasto asociado al elemento de oficina.
<b>Flujos alternativos</b>	

**Flujo alternativo 1a. El usuario introduce valores incorrectos.**

1. El sistema no permite registrar datos incorrectos.
2. El usuario introduce los datos correctamente y presiona la opción Registrar.
3. Volver al paso 4 del flujo básico.

**Pos-condiciones**

1. Se registra en el sistema el gasto asociado al elemento de oficina.

**Flujo alternativo 1b. El usuario introduce información incompleta.**

1. El sistema señala los datos vacíos y permite corregirlos.
2. El usuario corrige los datos y presiona la opción Registrar.
3. Volver al paso 4 del flujo básico.

**Pos-condiciones**

1. Se registra en el sistema el gasto asociado al elemento de oficina.

**Validaciones**

1. Se validan los datos según lo establecido en el Modelo conceptual.

<b>Conceptos</b>	N/A	N/A
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

**2.2.4 Validación de requisitos**

La validación de requisitos examina la especificación para asegurar que todos los requerimientos de *software* se han establecido de manera precisa; detectado las inconsistencias, omisiones y errores y que éstos han sido corregidos, y que los productos de trabajo cumplen con los estándares establecidos para el proceso, proyecto y producto (22). El objetivo de la validación de requerimientos es, ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento especificado para asegurarse que representan una descripción aceptable del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos (40).

Para la validación de los requisitos funcionales identificados se utilizó la técnica construcción de prototipos. Por cada requisito funcional se realizó la construcción de prototipos, de forma tal que se visualizaran los diferentes datos que tendría cada una de las interfaces. Además se realizó un acta de aceptación de los requisitos para el cliente. Los prototipos de interfaces de usuario se encuentran en el expediente de proyecto SIPAC 3.0 en el artefacto Especificación de requisitos.

**2.2.5 Requisitos no funcionales**

Los requisitos no funcionales son cualidades o propiedades que el producto final debe poseer. Con los requisitos no funcionales se especifican propiedades del sistema que tienen que ver con el

rendimiento, restricciones de ambiente y desarrollo, dependencias de plataforma y mantenimiento (41).

Los requisitos no funcionales que debe poseer el proceso de evaluación de gastos de funcionamiento interno fueron extraídos del expediente de proyecto SIPAC 2.0 del artefacto CIG-SPA-N-i3514-RNF.

### **Funcionalidad**

- Precisión

Breve descripción: Capacidad del *software* para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.

1. Los datos identificativos del lugar, operación o dato con error, mostrados en los mensajes de error del sistema, deberán ser 100% correctos.

### **Usabilidad**

- Operabilidad

Breve descripción: Capacidad del producto de *software* para permitirle al usuario operarlo y controlarlo.

1. La confirmación de la entrada de datos en cada formulario del sistema deberá poder hacerse mediante el uso de mouse o teclado.
2. El sistema permite búsqueda de los recursos materiales utilizando la denominación como criterio específico de búsqueda.

- Atracción

Breve descripción: Capacidad del producto de *software* de ser atractivo o amigable para el usuario.

1. El sistema diferenciará los mensajes de información de los mensajes de error y de advertencia valiéndose de distintos íconos para cada tipo.
2. El sistema presentará por cada interfaz, los términos capitalizados, es decir, la primera palabra tendrá su primera letra en mayúscula.
3. La tipografía y colores serán estándares en toda la aplicación.
4. En cada formulario del sistema los campos obligatorios serán señalados en rojo.

## 2.3 Modelado de la solución propuesta

### 2.3.1 Diagrama de clases del diseño

Los diagramas de clases del diseño describen la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de diseño de un sistema para crear el diseño conceptual de la información que se manejará en el mismo y los componentes que se encargarán de su funcionamiento así como la relación entre ellos. En las figuras que se muestran a continuación se presenta el diagrama de clases de diseño para la solución propuesta el cual está basado en el patrón arquitectónico MVC.

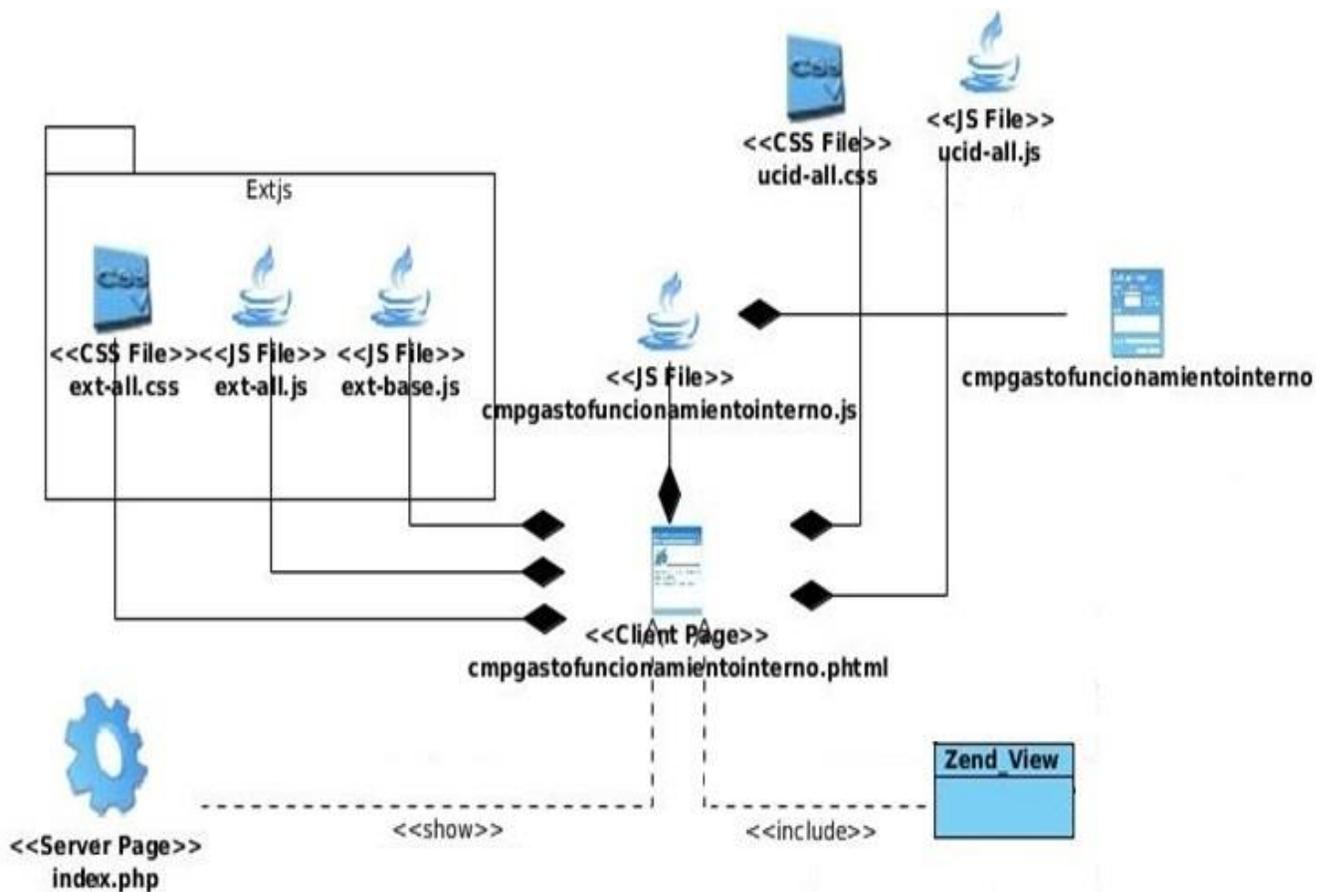


Figura 3: Vista del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar los gastos de las actividades de aseguramiento interno.



Figura 4: Controlador del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar el gasto de las actividades de aseguramiento interno.

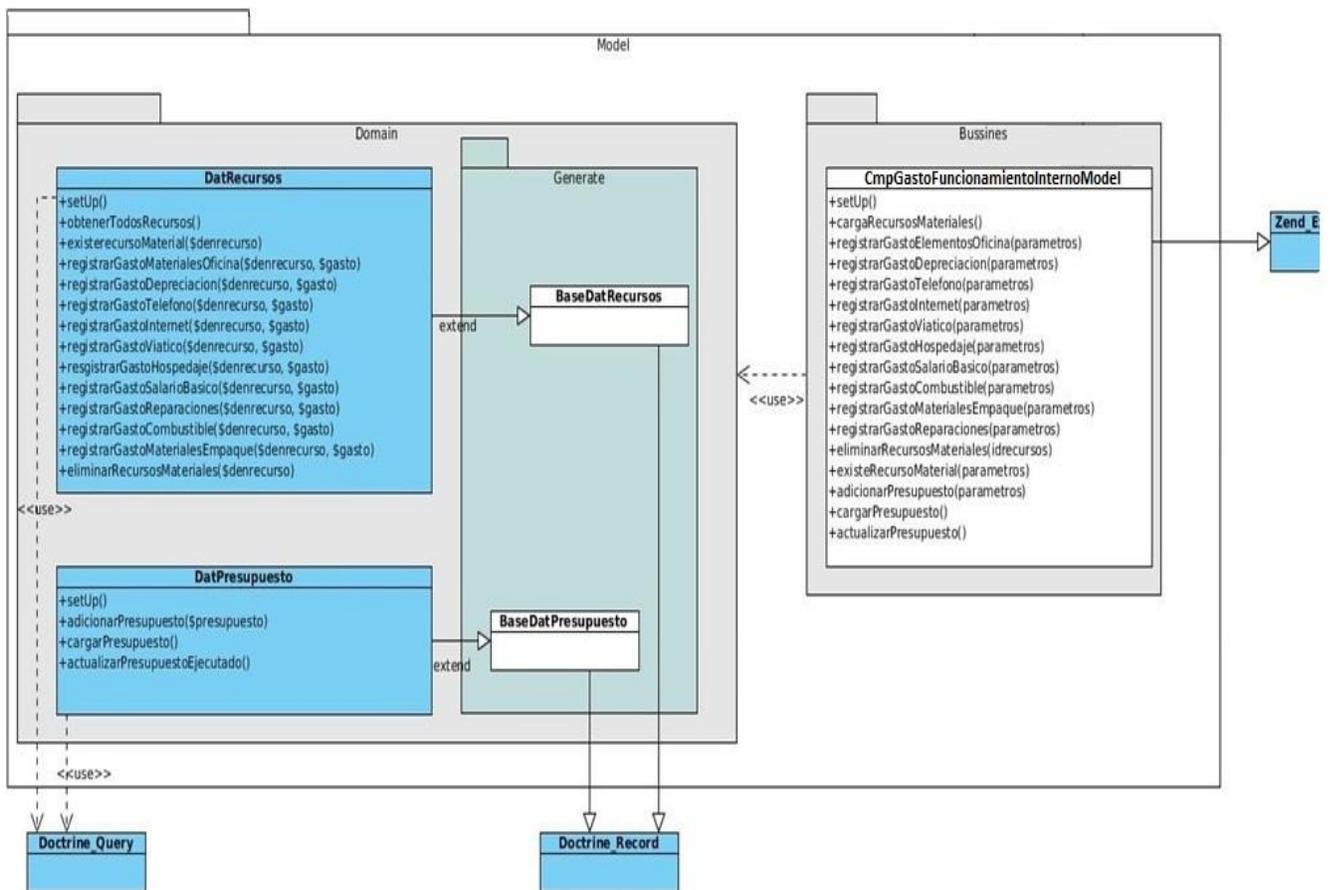
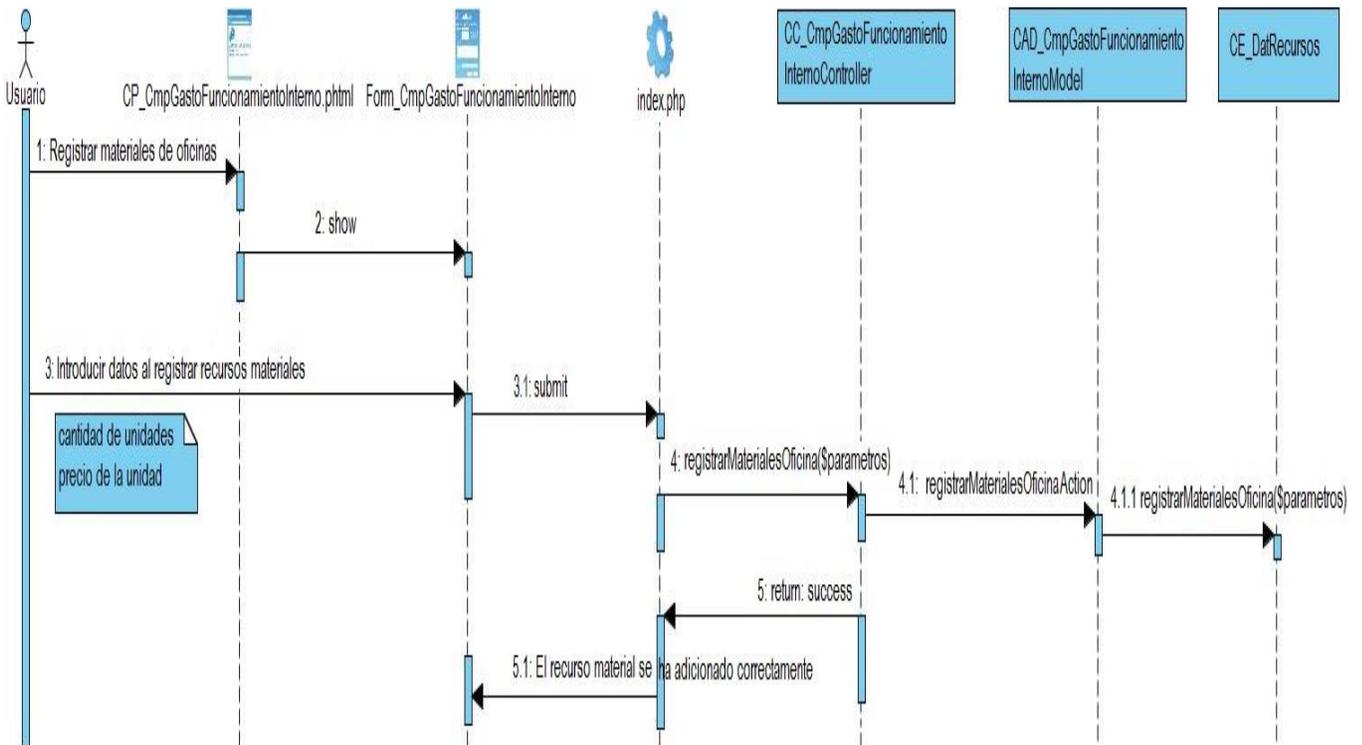


Figura 5: Modelo del patrón arquitectónico MVC del diagrama de clases de diseño para evaluar el gasto de las actividades de aseguramiento interno.

### 2.3.2 Diagrama de secuencia

Los diagramas de secuencias muestran la forma en que un grupo de objetos interactúan entre sí a lo largo del tiempo. Un diagrama de secuencia consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical (42). Permiten analizar detalladamente los procesos que se desarrollan mostrando la secuencia de mensajes entre objetos durante un escenario concreto. A continuación se presenta el diagrama de secuencia correspondiente al requisito funcional registrar gastos de los elementos de oficina.



**Figura 6: Diagrama de secuencia del requisito funcional Registrar gastos de los elementos de oficina.**

El usuario accede a la interfaz principal donde se muestran los recursos materiales para registrarle el gasto asociado a los mismos y selecciona la opción Registrar gastos de materiales de oficina. Se muestra un formulario donde el usuario debe introducir los datos del material de oficina que desea registrarle el gasto y presiona el botón Registrar para enviar los datos. La clase controladora CmpGastoFuncionamientoInternoController envía la petición a la clase modelo CmpGastoFuncionamientoInternoModel la cual realiza la consulta a la clase de acceso a datos DatRecursos y esta actualiza la tabla de los recursos materiales donde se registrarán los datos del

elemento de oficina registrado. Finalmente se muestra un mensaje señalando que el elemento de oficina fue registrado correctamente.

### **2.3.3 Modelo de datos**

Un modelo de datos es un conjunto de conceptos que pueden servir para describir la estructura de una base de datos. Está compuesto por entidades, atributos y relaciones. Por lo general los modelos de datos contienen un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la base de datos (43).

Se realizó un estudio al modelo de datos de SIPAC, donde se evidencia la necesidad de crear nuevas tablas para almacenar los datos referentes a los recursos materiales y el presupuesto para la evaluación de los gastos de las actividades de aseguramiento interno. Se incorporaron dos nuevas tablas `dat_recursos` y `dat_presupuesto`. Para el manejo de los datos referentes a las actividades y el plan, se utilizaron las entidades existentes en el modelo de datos de SIPAC, `dat_actividades` y `dat_plan_pdo`, además para establecer las relaciones entre el plan y las actividades como elementos de la planificación, se utilizaron las entidades `dat_elementos` y `dat_elementos.dat_elementos`.



Figura 7: Modelo de datos para la evaluación de las actividades de aseguramiento interno.

### 2.3.4 Diagrama de componentes

SIPAC se encarga de interrelacionar objetivos de trabajo y actividades en tiempo real de forma tal que se garantice el seguimiento del desarrollo y control del cumplimiento de los objetivos y tareas principales en las entidades. La arquitectura base de SIPAC presenta una estructura que responde a diferentes niveles de empaquetamiento: subsistema, componente, funcionalidad y requisito (2).

El componente planeación permite gestionar los elementos de la planificación (planes, objetivos, actividades, áreas de resultados clave, y factores que influyen en la planificación). En el componente planeación, la funcionalidad para la evaluación de los gastos de las actividades de aseguramiento interno se presenta como una funcionalidad horizontal relacionada con las funcionalidades verticales: Gestionar plan y Gestionar actividades, a las cuales se les realizaron las modificaciones asociadas a la solución, la **Figura 8** muestra el diagrama del componente planeación en SIPAC.

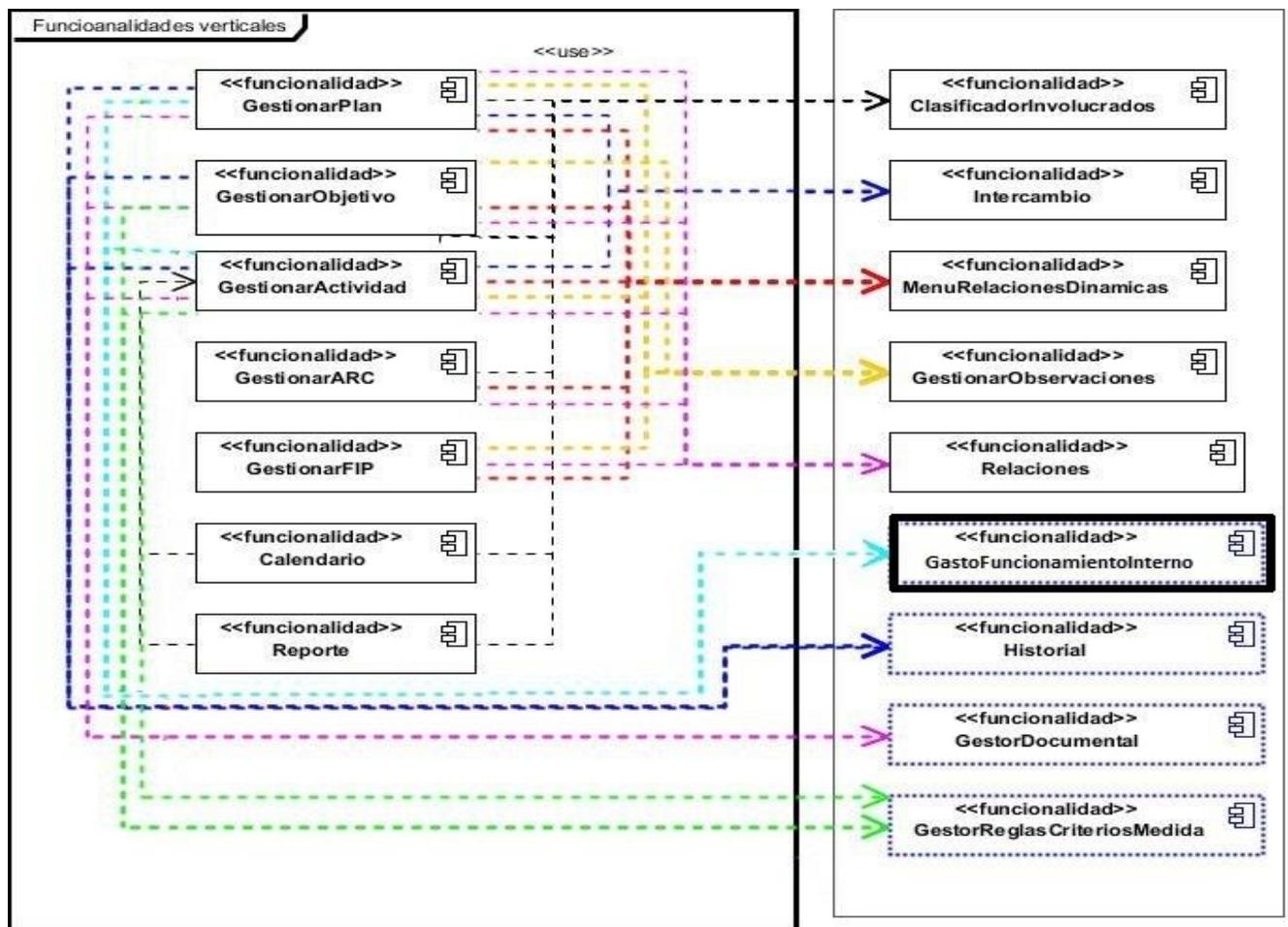


Figura 8: Diagrama del componente planeación en el SIPAC.

## 2.4 Patrones de diseño utilizados

### 2.4.1 Patrones GRASP

A continuación se presentan los patrones GRASP utilizados para el diseño de la solución.

**Experto:** La utilización del patrón experto se evidencia en la clase `CmpGastoFuncionamientoInternoModel`, encargada de gestionar los gastos de las actividades de aseguramiento interno.

**Creador:** La utilización del patrón creador se evidencia en la clase `CmpGastoFuncionamientoInternoController`, la cual crea instancias de los objetos de la clase `CmpGastoFuncionamientoInternoModel`.

**Bajo acoplamiento:** La utilización del patrón bajo acoplamiento se evidencia al definir las clases persistentes del modelo de datos.

**Alta cohesión:** El patrón alta cohesión fue utilizado en la solución de manera general, para que las clases del sistema no se afecten constantemente a los cambios y se puedan comprender mejor y reutilizar, tratando de no sobrecargar las clases con información que no esté relacionada con el contenido de la misma, además si la tarea encomendada es grande colabora con otros objetos para compartir el esfuerzo a realizar.

**Controlador:** La utilización del patrón controlador se evidencia en la clase controladora `CmpGastoFuncionamientoInternoController` que es la clase encargada de manejar los eventos de la solución propuesta.

### 2.4.2 Patrones GOF

Para el diseño de la solución se utilizaron los siguientes patrones GOF:

**Fachada:** En los sistemas basados en capas el patrón fachada define el punto de entrada a cada nivel ya que conoce los objetos del subsistema responsables para cada petición y delega en ellos las peticiones de los clientes. Este patrón fue utilizado en la clase `ZendExt_IoC` para acceder a los servicios de otros componentes dentro y fuera del subsistema.

**Cadena de Responsabilidad:** Este patrón fue utilizado para distribuir las responsabilidades en caso de ocurrencia de un error al realizarse una determinada consulta a la base de datos, ya que las clases modelos se encargan de realizar una excepción de tipo `ZendExt_Exception` la cual se envía a las clases controladoras y luego a la vista en un lenguaje entendible y la vista notifica el error al usuario mediante un mensaje.

### **Conclusiones parciales**

El desarrollo del presente capítulo permitió arribar a las siguientes conclusiones:

- Con la modelación de negocio se definieron los principales conceptos y procesos asociados a la implementación de la evaluación de los gastos de las actividades de aseguramiento interno, lo que permitió una mejor comprensión de la solución propuesta.
- Llevar a cabo la captura, descripción y validación de requisitos funcionales y no funcionales, permitió conocer los principales requerimientos que debe cumplir la solución propuesta.
- Se obtuvo el modelo de diseño de la solución como base para la implementación.
- Los patrones de diseño empleados permitieron especificar la estructura y comportamiento de las clases, garantizando los niveles de reutilización y mantenimiento de la solución propuesta.

## CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS

### Introducción

En el presente capítulo se realiza el modelo de implementación que da cumplimiento al diseño de la solución realizado en el capítulo anterior. Se describen los estándares de codificación definidos para la escritura del código, además se especifican un conjunto de validaciones y pruebas que evalúan la calidad del sistema desarrollado.

### 3.1 Validación del diseño propuesto

Las métricas definen la calidad durante el desarrollo del *software* a través de la medición de ciertas características internas. Esto permite al ingeniero evaluar la calidad durante el desarrollo del sistema (24).

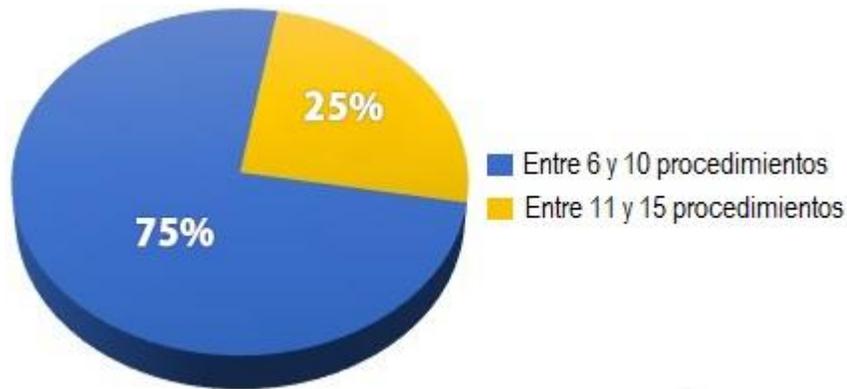
Para validar las clases del diseño se utilizaron las métricas: Tamaño operacional de clases (TOC) y Relación entre clases (RC). Estas métricas fueron diseñadas para validar los siguientes atributos de calidad:

- **Responsabilidad:** consiste en la responsabilidad asignada a una clase en un marco conceptual correspondiente al modelado de la solución propuesta.
- **Complejidad de implementación:** grado de dificultad que tiene implementar un diseño de clases determinado.
- **Reutilización:** grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de *software*.
- **Acoplamiento:** grado de dependencia o interconexión de una clase o estructura de clase con respecto a otras.
- **Complejidad del mantenimiento:** grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de *software*.
- **Cantidad de pruebas:** número o grado de esfuerzo para realizar las pruebas de calidad al producto diseñado.

**Métrica Tamaño operacional de clases:** se refiere al número de métodos asignados a una clase. Está determinada por los atributos: Responsabilidad, Complejidad de implementación y la Reutilización, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado. Para ver el modelo de medición utilizado en la métrica TOC ir al **Anexo 1**.

### Resultados del instrumento de evaluación de la métrica Tamaño operacional de clases

Representación en porcentaje de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

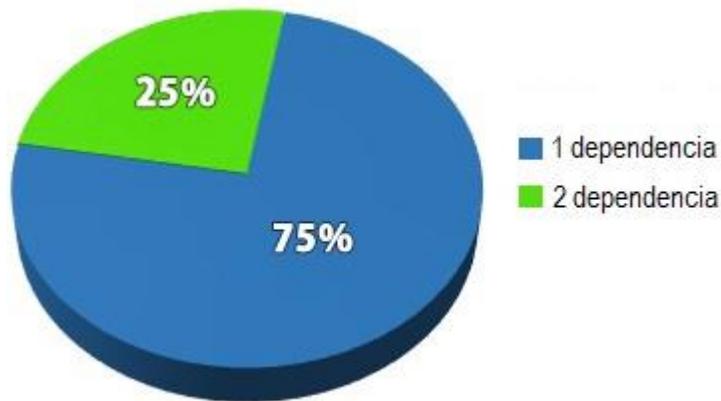


**Figura 9: Representación en porcentaje de la cantidad de clases y el número de procedimientos que contienen.**

Teniendo en cuenta que el 75% de las clases contienen un número de funcionalidades inferior a la media de procedimientos registrada en las mediciones que es de 10 procedimientos por clase, se concluye que el diseño propuesto para la evaluación de los gastos de las actividades de aseguramiento interno está entre los límites aceptables de calidad. Se observa cómo se fomenta la reutilización y cómo están reducidas en menor grado la responsabilidad y la complejidad de implementación.

**Métrica Relaciones entre clases:** dado por el número de relaciones de uso de una clase. Está determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado. Para ver el modelo de medición utilizado en la métrica RC ir al **Anexo 2**.

### Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)



**Figura 10: Representación en porcentaje de las asociaciones de uso por cantidad de clases.**

Luego de aplicar el instrumento de medición de la métrica RC, se concluye que el diseño propuesto para la evaluación de los gastos de las actividades de aseguramiento interno está entre los límites aceptables de calidad, teniendo en cuenta que el 75% de las clases poseen menos de 2 dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio; ya que el grado de acoplamiento de las clases es mínimo, la Complejidad de Mantenimiento, la Cantidad de Pruebas y la Reutilización se comportan favorablemente para un 75% de las clases.

## 3.2 Implementación de la solución propuesta

### 3.2.1 Estándares de codificación

Un estándar de codificación comprende todos los aspectos relacionados con la generación de código, de tal manera que sea prudente, práctico y entendible para todos los programadores. Para facilitar la comprensión y el mantenimiento de un *software*, el código fuente debe reflejar un estilo homogéneo y armonioso, como si un único programador lo hubiera escrito todo de una sola vez. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas (44).

Partiendo de lo dicho anteriormente, se definen 3 partes principales dentro de un estándar de programación:

- Convención de nomenclatura: define cómo nombrar variables, funciones y clases.
- Convenciones de legibilidad de código: forma de organizar el código y lograr que independientemente de quien desarrolle se entienda como un todo.

- Convenciones de documentación: define cómo establecer comentarios, archivos de ayuda, entre otros.

### **Nomenclatura de las clases**

Los nombres de las clases deben comenzar con mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing*, la cual define que los identificadores y nombres de variables, métodos y clases que están compuestos por múltiples palabras juntas, inicia cada palabra con letra mayúscula y sin usar ningún artículo posibilitando que con solo leer el nombre de la clase ya se reconozca la función de la misma.

### **Nomenclatura según el tipo de clases**

Clases controladoras: Las clases controladoras después del nombre llevan la palabra: “*Controller*”. Ejemplo: CmpGastoFuncionamientoInternoController.

Clases de los modelos:

*Business* (Negocio): Las clases que se encuentran dentro de *Business* después del nombre llevan la palabra: “*Model*”. Ejemplo: CmpGastoFuncionamientoInternoModel.

*Domain* (Dominio): Las clases que se encuentran dentro de *Domain* el nombre que reciben es el de la tabla en la base de datos. Ejemplo: DatRecursos.

*Generated* (Dominio base): Las clases que se encuentran dentro de *Generated* el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la base de datos. Ejemplo: BaseDatRecursos.

### **Nomenclatura de las funcionalidades**

El nombre a emplear para las funciones se escribe con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing* que es similar a la *PascalCasing* con la excepción de la primera letra.

Nomenclatura según la clase donde se encuentren las funciones:

- En la clase controladora: Las principales funcionalidades de las clases controladoras se les pone el nombre y seguida la palabra: “Action”. Ejemplo: eliminarRecursoMaterialAction ().
- En las clases de los modelos: Las funcionalidades se nombran de manera que al leerlo se identifique su propósito. Ejemplo de función: eliminarRecursoMaterial ().

### Nomenclatura de los comentarios

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

### 3.2.2 Evaluación de los gastos para las actividades de aseguramiento interno en SIPAC

La solución informática para evaluar los gastos para las actividades de aseguramiento interno en SIPAC, provee una interfaz amigable. A continuación se detallan los pasos a seguir para evaluar dicho proceso:

1-Se selecciona la opción Adicionar actividad, el sistema muestra una interfaz que contiene la pestaña recursos, mediante la cual se pueden registrar los recursos materiales que serán asociados a la actividad, ver **Figura 11**.

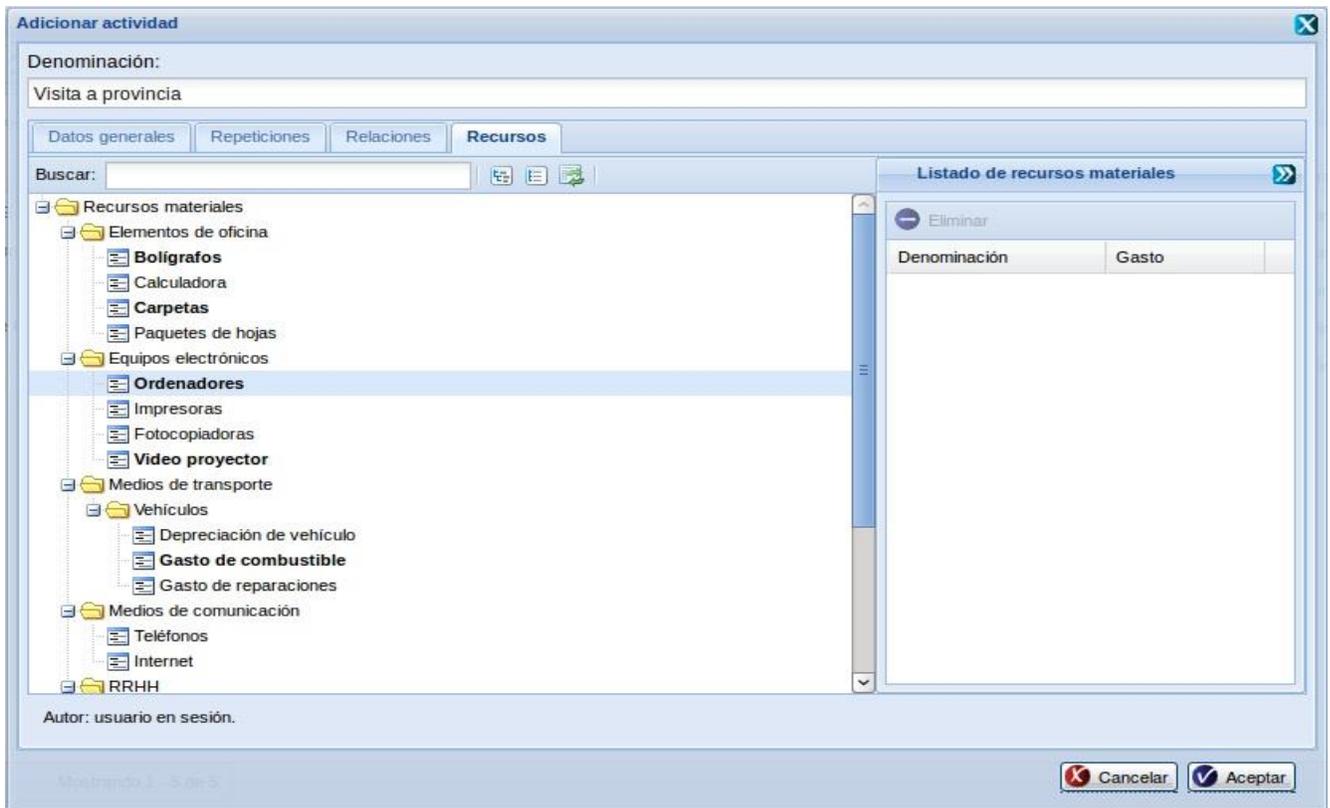
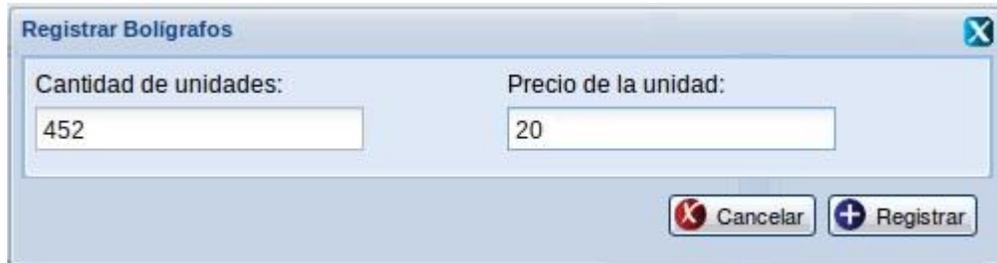


Figura 11: Adicionar actividad.

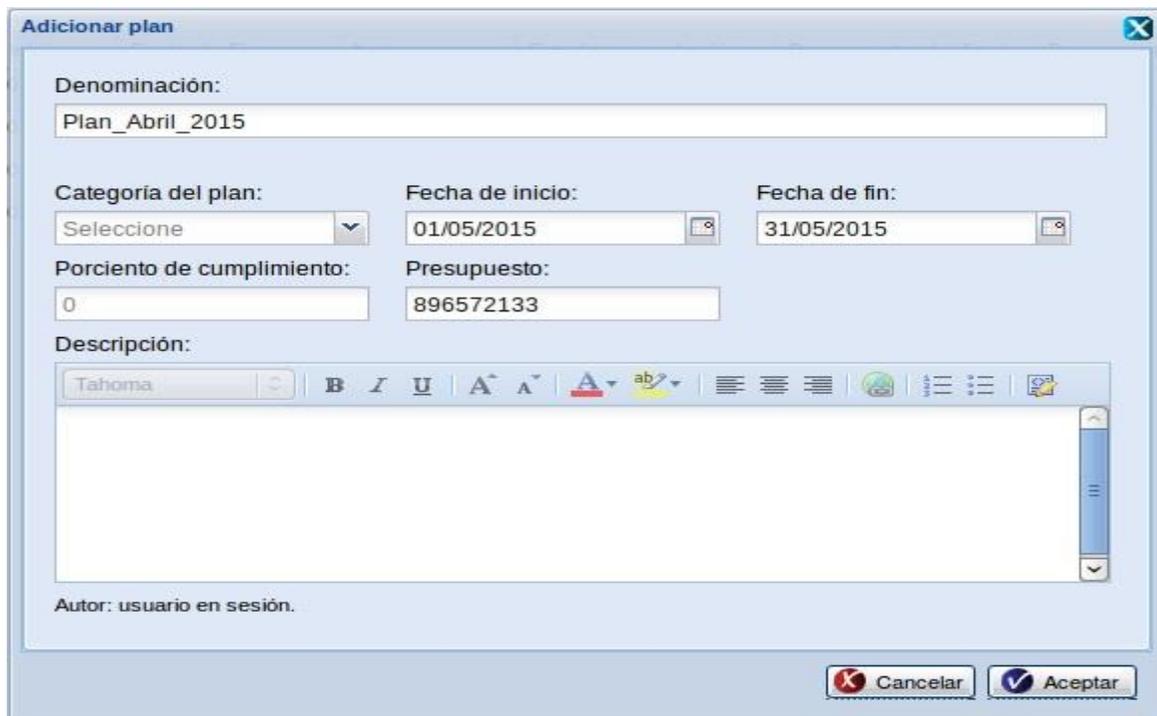
2-Se seleccionan los recursos materiales que serán asociados a la actividad, el sistema muestra una ventana con los campos a llenar para registrarle el gasto al recurso material seleccionado y muestra

en color negro los recursos adicionados a dicha actividad. La **Figura 12** muestra un ejemplo del registrar bolígrafos.



**Figura 12: Registrar gastos de bolígrafos.**

3- Se define el presupuesto planificado al adicionar el plan, ver **Figura 13** y se relacionan las actividades correspondientes a dicho plan.



**Figura 13: Definir presupuesto al plan.**

4- Se selecciona el plan y se presiona la opción Analizar plan, el sistema muestra una ventana con las actividades relacionadas a dicho plan y el gasto asociado a las mismas, además muestra el presupuesto planificado y el presupuesto ejecutado para el plan en cuestión, ver **Figura 14**.

Denominación	Gasto
Taller en el centro CEIGE	3390
Capacitar a los trabajadores	414
Viaje a provincia	1095
Capacitar estudiantes de la UCI	524

Figura 14: Analizar plan.

5- Al relacionar las actividades correspondientes al plan, se debe actualiza el presupuesto ejecutado al seleccionar la opción Analizar plan. Luego de actualizar el presupuesto ejecutado, el sistema muestra el presupuesto ejecutado en color verde si lo planificado es mayor que lo real ejecutado, sino muestra el presupuesto ejecutado en color rojo permitiendo que el usuario evalúe los gastos de funcionamiento interno, ver **Figura 15**.

CAR	Denominación	Fecha de Inicio	Fecha de Fin	Autor	Estado de aprobación	Presupuesto planificado	Presupuesto ejecutado	Estado de cumplimiento	Observ.	Anexos
9000003519	Plan Anual_2015	01 / 05 / 2015	31 / 05 / 2015	administrador	Creado	98659239874691	5423	Sin Cumplir (0%)		
9000003520	Plan Individual_2015	01 / 05 / 2015	31 / 05 / 2015	administrador	Creado	250	53100	Sin Cumplir (0%)		
9000003521	Plan Mensual_2015	01 / 05 / 2015	31 / 05 / 2015	administrador	Creado	1697	1619	Sin Cumplir (0%)		
9000003527	Plan Individual_Mayo_2015	01 / 05 / 2015	31 / 05 / 2015	administrador	Creado	412	1150	Sin Cumplir (0%)		

**Figura 15: Evaluación de los gastos de funcionamiento interno.**

El proceso de evaluación de los gastos para las actividades de aseguramiento interno en SIPAC, permite reducir el margen de error al calcular el costo de ejecución de cada actividad; proporcionando los métodos para calcular el gasto a los recursos asociados a una actividad, según lo que está establecido, y el gasto total de la misma. En el sistema se podrán realizar estos cálculos en menos de 5 segundos, lo que minimiza el tiempo en la planificación y evaluación del presupuesto. La solución permite que el usuario tenga control del registro de gastos, proporcionando un análisis del presupuesto de cada plan según las actividades asociadas. Se mantiene actualizado el registro de gastos internos en la organización para su funcionamiento. Por último la solución propuesta como funcionalidad agregada al SIPAC, posibilita el control y seguimiento de los recursos por los jefes en cada nivel de dirección.

### 3.3 Pruebas de *software*

Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un producto antes de su puesta en marcha. Básicamente, es una fase en el desarrollo de *software* que consiste en probar las aplicaciones construidas bajo condiciones especificadas, que tienen como objetivo encontrar errores y fallas para que el sistema cumpla con las especificaciones requeridas (45). Existen varios tipos de pruebas las cuales se describen a continuación.

**Pruebas unitarias:** se aplican a un componente o función del *software*. Su objetivo es aislar cada parte del programa y mostrar que las partes individuales son correctas. Tienen como propósito encontrar defectos en un módulo o función.

**Pruebas de integración:** consiste en construir el sistema a partir de los distintos componentes y probarlo con todos integrados. Su propósito es asegurar que no existan errores de interfaces y encontrar defectos en el sistema.

**Pruebas de aceptación:** son las únicas pruebas que son realizadas por los usuarios expertos, todas las anteriores las lleva a cabo el equipo de desarrollo. Consiste en comprobar si el producto está listo para ser implantado para el uso operativo en el entorno del usuario.

**Pruebas del sistema:** son usualmente conducidas para asegurar que todos los módulos trabajan como sistema sin error. Es similar a la prueba de integración pero con un alcance mucho más amplio.

#### Métodos de prueba

Los métodos de prueba del *software* tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Ayudan a definir conjuntos de casos de prueba aplicando ciertos criterios. Los métodos de prueba que se especifican a continuación permiten probar cada una de las condiciones existentes en el programa, identificar claramente las entradas, salidas y estudiar las relaciones que existen entre ellas, permitiendo así maximizar la calidad de las pruebas y de este modo obtener un sistema más estable y confiable.

#### ➤ Pruebas de caja blanca

También suelen ser llamadas estructurales o de cobertura lógica. Con ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba que ejecutan cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones

manejadas por un programa (2). Este tipo de prueba le permite al ingeniero de *software* obtener casos de pruebas que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdaderas y falsas.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Dentro de la prueba de caja blanca se incluyen las técnicas de pruebas que serán descritas a continuación:

Prueba del camino básico: permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto de caminos básicos.

Prueba de condición: ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.

Prueba de flujo de datos: se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.

Prueba de bucles: se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución todos los bucles en sus límites operacionales.

### **Aplicación de las pruebas de caja blanca**

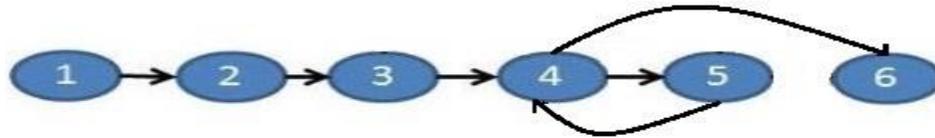
Con el objetivo de valorar la calidad con la que se llevó a cabo la implementación de la solución propuesta, fue necesario aplicar una de las técnicas descritas anteriormente, en este caso la del camino básico.

Para ello fue necesario seguir los siguientes pasos básicos:

1. A partir del diseño o del código fuente, dibujar el grafo de flujo asociado.
2. Calcular la complejidad ciclomática del grafo.
3. Determinar un conjunto básico de caminos independientes.
4. Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Dando cumplimiento a los anteriores pasos básicos se enumeran cada una de las sentencias de código de uno de los procedimientos de la clase `CmpgastofuncionamientointernoModel`, específicamente la funcionalidad `obtenerRecursosSegunActividad`. Ver **Anexo 3**.

Luego de realizado el procedimiento del primer paso, se hace necesario representar el grafo de flujo asociado al código de la funcionalidad presentada anteriormente. Para ello se utilizan aristas, nodos y regiones.



**Figura 16: Grafo de flujo asociado al procedimiento `obtenerRecursosSegunActividad`.**

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres vías o fórmulas de manera tal que quede justificado el resultado, siendo el mismo en cada caso:

1.  $V(G) = (A - N) + 2$

**Resultado:**  $V(G) = (6 - 6) + 2 = 2$

2.  $V(G) = P + 1$

**Resultado:**  $V(G) = 1 + 1 = 2$

3.  $V(G) = R$

**Resultado:**  $V(G) = 2$

A: cantidad total de aristas del grafo.

N: cantidad total de nodos del grafo.

P: cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

R: cantidad total de regiones existentes en el grafo, se incluye el área exterior del grafo como una región más.

El cálculo efectuado mediante las fórmulas antes presentadas muestra una complejidad ciclomática de valor 2, de manera que existen dos posibles caminos por donde el flujo puede circular, este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

**Camino básico #1:** 1 – 2 – 3 – 4 – 5 – 4 – 6

**Camino básico #2:** 1 – 2 – 3 – 4 – 6

A continuación se describen los casos de prueba para cada uno de los caminos básicos determinados en el grafo de flujo:

**Tabla 5: Caso de prueba para el camino básico # 1.**

<b>Camino básico #1: 1 – 2 – 3 – 4 – 5 – 4 –6</b>	
Descripción	Se cargan los recursos asociados a la actividad.
Condición de ejecución	La actividad ha sido seleccionada y tiene recursos materiales asociados a la misma.
Resultado esperado	Teniendo en cuenta el identificador de la actividad seleccionada se cargan los recursos materiales correspondientes a la misma.

Para ver el caso de prueba del camino básico # 2 ir al **Anexo 4**.

Luego de aplicar los casos de prueba correspondientes a cada camino básico identificado, se comprobó el correcto funcionamiento del procedimiento obtenerRecursosSegunActividad.

➤ **Pruebas de caja negra**

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del *software*. Estas pruebas se limitan a brindar solo datos como entradas y estudiar las salidas, sin preocuparse de lo que pueda estar haciendo el componente internamente, es decir, solo trabaja sobre su interfaz externa (46).

Estas pruebas permiten encontrar:

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Entre las técnicas para desarrollar las pruebas de caja negra se encuentran:

Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.

Análisis de valores límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Grafos de causa-efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

### Aplicación de las pruebas de caja negra

A continuación se aplica la técnica Partición de equivalencia como parte de la realización de la prueba de caja negra sobre la interfaz que responde al requisito funcional Registrar gastos de elementos de oficina. Para ello se definieron variables de equivalencia que representan un conjunto de estados válidos (representan entradas válidas al programa) y no válidos (representan valores de entradas erróneas) para las condiciones de entradas del sistema.

**Tabla 6: Caso de prueba de caja negra para validar el requisito funcional Registrar gastos de elementos de oficina**

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Registrar gastos de elementos de oficina.	1. El sistema debe permitir que el usuario al seleccionar el elemento de oficina, adicione un nuevo elemento de oficina con el gasto calculado.	EP 1.1: Flujo básico de eventos.	1. Se selecciona la opción <b>Inicio -Planificación - Actividades -Adicionar - Recursos.</b>  2. Se selecciona el elemento de oficina que se desea registrar, se introducen los datos y se presiona el botón <b>Registrar.</b>  3. Se adiciona el elemento de oficina seleccionado.
		EP 1.2: Se introducen valores incorrectos.	1. Se selecciona la opción <b>Inicio -Planeación -</b>

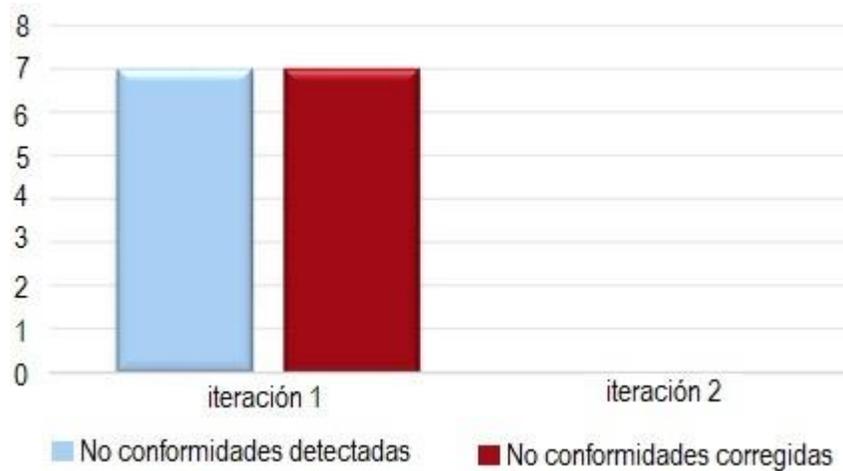
			<p><b>Actividades -Adicionar – Recursos.</b></p> <p>2. Se selecciona el elemento de oficina que se desea registrar, se introducen datos que no son números y se presiona el botón <b>Registrar</b>.</p> <p>3. El sistema no permite al usuario introducir caracteres diferentes de números.</p>
		<p>EP 1.3: Se introduce información incompleta.</p>	<p>1. Se selecciona la opción <b>Inicio -Planificación - Actividades -Adicionar – Recursos.</b></p> <p>2. Se selecciona el elemento de oficina que se desea registrar, se presiona el botón <b>Registrar</b> dejando campos en blanco.</p> <p>3. El sistema muestra el mensaje de error: “Por favor verifique que hay campo(s) vacío(s)”.</p>

Para ver los juegos de datos a probar ir al **Anexo 5**.

### 3.4 Resultados de las pruebas aplicadas

Luego de aplicar los métodos de prueba a las funcionalidades implementadas, se obtuvieron resultados satisfactorios desde el punto de vista interno y funcional, atendiendo al correcto comportamiento del mismo ante diferentes situaciones. Las no conformidades detectadas fueron debidamente atendidas en aras de lograr el correcto funcionamiento de la solución desarrollada. A

continuación se muestra la cantidad de no conformidades detectadas en cada iteración de pruebas realizadas:



**Figura 17: No conformidades detectadas durante las iteraciones de pruebas realizadas.**

### Conclusiones del capítulo

El desarrollo del presente capítulo permitió arribar a las siguientes conclusiones parciales:

- La utilización de estándares de codificación, así como los prototipos de las interfaces con los que podrá interactuar el usuario, permitió definir la estructura física y lógica de la solución.
- La aplicación de las métricas Tamaño operacional de clases y Relaciones entre clases, permitió validar el diseño propuesto donde se evidencia que la propuesta de solución se encuentra entre los límites aceptables de calidad.
- La aplicación de pruebas de caja blanca (Técnica del camino básico) y pruebas de caja negra (Técnica partición de equivalencia) posibilitó verificar el correcto funcionamiento de la evaluación de los gastos de funcionamiento interno.

## CONCLUSIONES

La realización del presente trabajo y los resultados obtenidos con el mismo permitieron arribar a las siguientes conclusiones generales.

1- Se elaboró el marco teórico de la investigación donde se realizó un estudio de varios sistemas de planificación que incluyen el proceso de evaluación de gastos de funcionamiento interno, lo cual arrojó, que no existe un sistema que integre el proceso de evaluación de gastos de funcionamiento interno y se evidencia la necesidad de crear el mismo.

2- Se realizó el diseño y la implementación de la evaluación de los gastos para las actividades de aseguramiento interno en SIPAC, con el objetivo de llevar un mejor registro, seguimiento y control de los recursos en la planeación operativa, mediante la asignación de recursos a actividades de aseguramiento interno para evaluar la ejecución de las mismas y controlar lo planificado con lo real ejecutado, permitiendo tomar mejores decisiones.

3- Se evaluó la viabilidad de la solución a través de pruebas de *software* efectuadas para el nivel de unidad, las cuales arrojaron resultados favorables posibilitando dar cumplimiento a las funcionalidades previstas para el mismo.

## **RECOMENDACIONES**

- Debido que cada entidad maneja gastos propios para realizar sus actividades, se recomienda definir nomencladores que tributen a la especificación de los gastos de funcionamiento de cada entidad.
- Se recomienda incluir en los reportes del SIPAC la evaluación del presupuesto al plan.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Castro, Raúl.** Instrucción no. 1 Del Presidente de Los Consejos de Estado y de Ministros para la Planificación de los Objetivos y Actividades en los Órganos, Organismos de La Administración Central del Estado, Entidades Nacionales y las Administraciones locales del Poder Popular. Habana : s.n., 2012.
2. **Santamaría, Lisandra y Rodríguez, Alejandro.** Gestor de reglas y criterios de medidas para evaluar el cumplimiento de objetivos estratégicos en el Sistema de Planificación de Actividades (SIPAC). Habana : s.n., 2014.
3. **Valiente, Lucrecia de la Caridad y León, Juana Dailiana.** Módulo Configuración del Sistema para la Planificación de Actividades (SIPAC). Habana : s.n., 2012.
4. **MINISTERIO DE FINANZAS Y PRECIOS.** RESOLUCIÓN No.235-2005.
5. **Definiciones miles de términos explicados, funcionamiento.** [En línea] [Citado el: 14 de Marzo de 2015] <http://definicion.mx/?s=Definici%C3%B3n+de+funcionamiento&submit=Search>.
6. **Definición .DE, Concepto de interno.** [En línea] [Citado el: 14 de Marzo de 2015] <http://definicion.de/interno/>.
7. **Definición .DE, Recursos materiales.** [En línea] [Citado el: 10 de Marzo de 2015] <http://definicion.de/recursos-materiales/>.
8. **Definición miles de términos explicados.** [En línea] [Citado el: 10 de Marzo de 2015] <http://definicion.mx/recursos-materiales/>.
9. **Definiciones miles de términos explicados, RRHH.** [En línea] [Citado el: 14 de Marzo de 2015.] <http://definicion.mx/recursos-humanos/>.
10. **Definición .DE, Equipos de oficina.** [En línea] [Citado el: 14 de Marzo de 2015] <http://definicion.de/equipo-de-oficina/>.
11. **Consumoteca, Equipos electrónicos.** [En línea] [Citado el: 14 de Marzo de 2015] <http://www.consumoteca.com/electrodomesticos/gestion-de-residuos/aparato-electrico-y-electronico/>.
12. **Buenas tareas, Materiales de empaque.** [En línea] [Citado el: 14 de Marzo de 2015] <http://www.buenastareas.com/materias/concepto-de-material-de-empaque/0>.

13. **Definicion .DE, Medios de transporte.** [En línea] [Citado el: 14 de Marzo de 2015]  
<http://definicion.de/transporte/>.
14. **Biblioteca virtual.** [En línea] [Citado el: 14 de Marzo de 2015]  
<http://www.banrepcultural.org/blaavirtual/ayudadetareas/periodismo/losmediosdecomunicacion>.
15. **Significado, Presupuesto.** [En línea] [Citado el: 14 de Marzo de 2015]  
<http://www.significados.com/presupuesto/>.
16. **Definición abc.** [En línea] [Citado el: 10 de Marzo de 2015]  
<http://www.definicionabc.com/general/recursos-materiales.php>.
17. **SLIDESHARE.** [En línea] [Citado el: 8 de Abril de 2015]  
<http://es.slideshare.net/julianamonsalveg/administracion-de-proyectos-30725176?related=3>.
18. **Empresa&Economía.** [En línea] [Citado el: 8 de Abril de 2015]  
<http://www.empresayeconomia.es/aplicaciones-para-empresas/openproj-herramienta-para-el-diseno-libre-de-proyectos.html>.
19. **Lemus, Jorge y Navas, Jennifer.** Manual de OpenProj. Colombia : s.n., 2009.
20. **Ecured.** GESPRO. [En línea] [Citado el: 8 de 5 de 2015] <http://www.ecured.cu/index.php/Gespro>.
21. **Colonia, Ricardo.** Manual Microsoft Project 2007. [En línea] 2011. [Citado el: 8 de Abril de 2015].  
<http://es.slideshare.net/rcolonia/manual-microsoft-project-2007-espaol?related=1>.
22. **Pressman, Roger.** Ingeniería de software. Un enfoque práctico 6ta Edición. Nueva York, E.U.A : McGraw-Hill, 2007.
23. **Larman, Craig.** UML y Patrones Introducción al análisis y diseño orientado a objetos. México : Prentice Hall, 1999.
24. **Trujillo, Yadelis y Fernández, Radamés.** Subsistema Mensajería SWIFT Fase II. Habana : s.n., 2013.
25. **Fernández, Mairelys y Zorrilla, Osley.** Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Las Entidades CEDRUX. Habana : s.n., 2010.

26. **Escuela de Ciencias y Sistemas Universidad de San Carlos de Guatemala.** UML (Unified Modeling Language). [En línea] [Citado el: 27 de Noviembre de 2014] <http://www.scribd.com/doc/2080534/UML>.
27. **BPMN (Business Process Modeling Notation).** [En línea] [Citado el: 27 de Noviembre de 2014.] [http://www.ecured.cu/index.php/Business\\_Process\\_Modeling\\_Notation](http://www.ecured.cu/index.php/Business_Process_Modeling_Notation).
28. **INTRODUCCION A HTML.** [En línea] [Citado el: 27 de Noviembre de 2014] <http://eusalud.uninet.edu/Cursos/doc99/INTROHTML.html>.
29. **Tutorial de CSS.** [En línea] [Citado el: 1 de Diciembre de 2014] <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>.
30. **JavaScript.** [En línea] [Citado el: 1 de Diciembre de 2014] <http://www.desarrolloweb.com/javascript/>.
31. **González, Yankiel.** Solución para la generación dinámica de componentes en el Marco de Trabajo Sauxe. Habana : s.n., 2012.
32. **Pérez, Liu y Vázquez, Jiorqui.** Componente Gestor de Intercambio de Información JIT para el Sistema para la Planificación de Actividades (SIPAC). Habana : s.n., 2012.
33. **Introducción a AJAX.** [En línea] [Citado el: 27 de Noviembre de 2014] [http://librosweb.es/ajax/capitulo\\_1.html](http://librosweb.es/ajax/capitulo_1.html).
34. **Torres, Odelsis.** Desarrollo de la versión 1.1 del subsistema Costos y Procesos del Sistema Integral de Gestión CEDRUX. Habana : s.n.
35. **Firefoxmania.** [En línea] [Citado el: 27 de Noviembre de 2014] <http://firefoxmania.uci.cu/download/?p=faq&a=fx>.
36. **Firefox.** [En línea] [Citado el: 27 de Noviembre de 2014] <http://conceptodefinicion.de/firefox/>.
37. **Laboratorio nacional de calidad del software.** Ingeniería del software: Metodologías y ciclos de vida. España : s.n., 2009.
38. **Di Biase, Franco y Di Biase, Aldo.** Sistema de información computacional.
39. **Sommerville, Ian.** Ingeniería del software séptima edición. Madrid : s.n., 2005.
40. **Arias, Michael.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. 2006.

41. **Alfonso, Inna.** Desarrollo del componente para la configuración visual de las excepciones en el marco de trabajo Sauxe. La Habana : s.n., 2011.
42. **Gutierrez, Demián.** UML Diagrama de Secuencia. Los Andes : s.n., 2011.
43. **Laurens, Yenifer.** Modelo Conceptual de datos.
44. **Becerra, Fernando.** [En línea] [Citado el:2 de Mayo de 2015] <http://serk.kualtus.com/codigo.htm>.
45. **Ruiz, Roberto.** Las pruebas de software y su importancia en las organizaciones. Veracruz : s.n., 2011.

## **BIBLIOGRAFÍA CONSULTADA**

- 1- **Pedraza, Lina O.** RESOLUCIÓN no.354 del Ministerios de Finanzas y Precios. 2013.
- 2- **Castro, Raúl.** Instrucción no.1 Del Presidente de Los Consejos de Estado y de Ministros para la Planificación de los Objetivos y Actividades en los Órganos, Organismos de La Administración Central del Estado, Entidades Nacionales y las Administraciones Locales. del Poder Popular. Habana : s.n., 2012.
- 3- **Pressman, Roger.** Ingeniería de software. Un enfoque práctico 6ta Edición. Nueva York, E.U.A : McGraw-Hill, 2007.
- 4- **Larman, Craing.** UML y Patrones Introducción al análisis y diseño orientado a objetos. México : Prentice Hall, 1999.
- 5- **Ribas, Gretel.** Procedimiento de estimación de tiempo y esfuerzo del proceso de despliegue del sistema Cedrux. La Habana : s.n., 2011
- 6- **Sommerville, Ian.** Ingeniería del software séptima edición. Madrid : s.n., 2005.
- 7- **Rendón, Ariatna.** Aquitectura base de SIPAC. La Habana: s.n., 2009.
- 8- **Pérez, Liu y Vázquez, Jiorqui.** Componente Gestor de Intercambio de Información JIT para el Sistema para la Planificación de Actividades (SIPAC). Habana : s.n., 2012.
- 9- **Valiente, Lucrecia de la Caridad y León, Juana Dailiana.** Módulo Configuración del Sistema para la Planificación de Actividades (SIPAC). Habana : s.n., 2012.
- 10- **Arias, Michael.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. 2006.
- 11- **Universidad de las Ciencias Informáticas (UCI).** Metodología de desarrollo para la Actividad productiva de la UCI . Habana : s.n., 2014.

## **GLOSARIOS DE TÉRMINOS**

**Código abierto (en inglés Open Source):** término con el que se conoce al *software* distribuido y desarrollado libremente.

**Software libre:** es la denominación del *software* que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.

**Navegador:** *software* que permite al usuario recuperar y visualizar documentos de hipertexto desde servidores web a través de Internet.

**Herramientas CASE (Computer Aided Software Engineering, 'Ingeniería de Software Asistida por Computadora'):** son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software*.

**Framework:** es una estructura de soporte definida, mediante la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**XML (Extensible Markup Language, 'Lenguaje de marcas extensible'):** es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

**IEEE (Institute of Electrical and Electronics Engineers, 'Instituto de Ingeniería Eléctrica y Electrónica'):** es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

## ANEXOS

**Anexo 1:** Material de medición de la métrica Tamaño operacional de clases (TOC).

**Tabla 7: Material de medición de la métrica TOC.**

Características	Categoría	Criterio
Responsabilidad	Baja	$\leq \text{Promedio(Prom)}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$> 2*\text{Prom}$
Complejidad de implementación	Baja	$\leq \text{Prom}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$> 2*\text{Prom}$
Reutilización	Baja	$> 2*\text{Prom}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$\leq \text{Prom}$

**Anexo 2:** Material de medición de la métrica Relaciones entre clases (RC).

**Tabla 8: Material de medición de la métrica RC.**

Características	Categoría	Criterio
Acoplamiento	Ninguno	0 dependencia
	Bajo	1 dependencia
	Medio	2 dependencia
	Alto	$> 2$ dependencia
Complejidad de mantenimiento	Baja	$\leq \text{Promedio(Prom)}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$> 2*\text{Prom}$
Reutilización	Baja	$> 2*\text{Prom}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$\leq \text{Prom}$
Cantidad de pruebas	Baja	$\leq \text{Prom}$
	Media	Entre Prom y $2*\text{Prom}$
	Alta	$> 2*\text{Prom}$

**Anexo 3:** Fragmento de código con el algoritmo obtenerRecursosSegunActividad.

```

/*Función para cargar los recursos de la actividad pasada por parámetro*/
public function obtenerRecursosSegunActividad($idactividad) {
    $obj_den_recursos = new DatRecursos(); //1
    if ($idactividad) { //2
        $result = $obj_den_recursos->getTodos($idactividad); //3
        $recursos = array(); //3
        foreach ($result as $value) { //4
            $recursos[] = $value; //5
        }
        return $recursos; //6
    }
}

```

**Figura 18:** Fragmento de código del algoritmo obtenerRecursosSegunActividad.

**Anexo 4:** Caso de prueba para el camino básico # 2.

**Tabla 9:** Caso de prueba para el camino básico # 2.

Camino básico #2: 1 – 2 – 3 – 4 – 6.	
Descripción	Se cargan los recursos asociados a la actividad.
Condición de ejecución	La actividad seleccionada no tiene recursos materiales asociados a la misma.
Resultados esperados:	No se cargan recursos materiales para la actividad seleccionada.

**Anexo 5:** Juego de datos a probar para el caso de prueba del requisito funcional Registrar gastos de elementos de oficina.

**Tabla 10:** Juego de datos a probar para el caso de prueba del requisito funcional Registrar gastos de elementos de oficina.

Id del escenario	Escenario	Cantidad de unidades	Precio de la unidad	Respuesta del sistema
EP1.1:	Registrar gastos de elementos de oficina introduciendo datos válidos.	V(9875651116)	V(650122555)	El sistema actualiza el listado de los recursos materiales con el gasto calculado.
		V(10)	V(1)	
EP 1.2:	Registrar gastos de elementos de oficina introduciendo información incompleta.	(letras)	(*/-)	El sistema no permite al usuario introducir caracteres diferentes de números.
EP 1.3	Registrar gastos de elementos de oficina dejando campos en blanco.	V(Vacío)	V(Vacío)	El sistema muestra el mensaje de error: “Por favor verifique que hay campo(s) vacío(s)”.
EP 1.4	Cancelar.	N/A	N/A	Se cancela la operación y se cierra la ventana.